إعدادات التطبيق **Application Settings**

تاريخ مقتضب عن الإعدادات A Short History of Settings

لقد بحث المبرمجين عن طريقة مريحة لصيانة القيم القابلة للتركيب في تطبيقاتهم.في الأيام الأولى للتطوير بميكروسوفت دوسMS-DOS ،كان التركيب متاح للجميع،كل برنامج يوفر نظام إعداداته الخاصة.العديد من التطبيقات لا تحتاج تركيب متخصص،ولكن تلك التي تخزن إعدادات التركيب مع بيانات التطبيق المدارة،جميعها في ملفات .dat خاصة.

مع حلول تطوير ويندوز السائد،قدمت ميكروسوفت ملف معتمد على إدارة الإعدادات من خلال "واجهة البرمجة التطبيقية *application programming interface*(API)" الخاصة به."المعنون الخاص API" يستدعي GetPrivateProfileInt) ،GetPrivateProfileString،SetPrivateProfileString ،إلخ) والتي تزود طريقة قياسية لتخزين قيم تركيب مختصرة في تنسيق ملف نصي مفتوح و سهل الفهم.استخدمت ميكروسوفت الملفات" INI " (مسماة من أجل امتداد ملفها *.ini*) هذه من أجل التركيب الخاص بها.العديد من هذه الملفات تبقى قابعة في مجلد ويندوز *Windows*لنظامك.إليك المحتوى الموجود في ملف نظامي *win.ini* .

; for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[Mail]
MAPI=1
CMCDLLNAME32=mapi32.dll
CMC=1
MAPIX=1
MAPIXVER=1.0.0.1
OLEMessaging=1
[MCI Extensions.BAK]
aif=MPEGVideo

التنسيق لملف INI سهل الفهم.كل ملف متضمن مقاطع مسماة معرفة ضمن أقواس مربعة،كما في [الخطوط fonts].كل مقطع محفوظ كمجموعة أزواج : قيمة - مفتاح في نموذج "المفتاح key = القيمةvalue ."التنسيق بسيط بشكل كافي بحيث يستطيع أياً كان استخدام المفكرة Notepadلعمل تغيرات.وليس حتى من الصعب بالنسبة لبرنامج من أن يكتب إجراءاته الروتينية الإدارية لملف INI-file الخاص به،ولكن تضمينهم في واجهة برمجة تطبيقية لويندوز Windows API يجعل منها أكثر جاذبية.ومن ثم جاءت الفوضى.مع اختيار العديد من البرامج تخزين ملفات التركيب الخاصة بها فيما يعرف بالموقع المركزي،فإن مجلد *Windows*أصبح بسرعة الملف المكافئ لمحطة غراند المركزية يوم الجمعة عند الساعة الخامسة مساءاً في نيويورك.كانت السرعة هي المشكلة،وأيضاً بما أن الثبات في تفسير وإعادة كتابة ملفاتINI تستهلك مصادر وحدة المعالجة المركزية CPU الثمينة.

تقدمت ميكروسوفت بحل:التسجيل*registry* .فهو عبارة عن قاعدة بيانات هرمية من أزواج قيمة مفتاح keyvalue تنظف نظام الملفات filesystemوتجلب تحسينات السرعة لإدارة الإعداد.وهو يضيف أيضاً إدارة إعدادات الأمان مسبقة التعريف جديدة لإمكانية الوصول للمسجلregistry ،ويوفر دعم لبعض البيانات المحددة النوع.

ولكن ميزة واجهة البرمجة التطبيقية (أو واجهة برمجة التطبيقAPI)الجديدة لم تكن سهلة الاستخدام والفهم(على الرغم من أن الفيجوال بيسك عملت على تضمين أوامر بسيطة،مثل GetSetting والتي توفر إمكانية وصول محدودة لمفاتيح التسجيل registry keys والقيمvalues .)

تقنية إلقاء التسجيل (أو طرح التسجيل) عند وجود مشكلة في قيم الإعداد،ولكن هذه التقنية لم تكن نصراً كاملاً.فمع العديد من البائعين الذين يحشون الكثير من البيانات في الريجستري،يصبح الانتفاخ(التضخم) مشكلة مرةً أخرى.ومع إدارة النظام فإن الجميع بإمكانه الوصول إلى الريجستري،وبالتالي كلما تضخم الريجستري،كلما أصبح الأداء أسوء.

تضمن الإصدار الأول للدوت نت ملفات إعداد خاصة بالتطبيق،تعود نوعاً ما إلى أيام ملف INI-file.في بعض الطرق،ملفات app.config و web.config كانت أفضل من ملفات INIبما أنها تحتوي محتوى تركيب XML.ولكن ملفات INI لديها تركيب،وتستطيع تحديثها بالمفكرة. ملفات config للدوت نت كانت ذات صيت سيء فيما يخص صعوبة التحديث،إما ضمن تطبيق الدوت نت أو خارجياً في المفكرة(تبعاً لإصدارات الذاكرة المخفية السحريةweird caching) وأيضاً ملفات*config*. ليس لديها أمن securityولا حتى تحديد أنواع بيانات قوية strong data typing متاحة في الريجستري.

لقد دفعت إعدادات التركيب على الأقل مستوى ما من القلق لدى المبرمجين منذ الظهور الأول للويندوز.ولكن نظام إعدادات جديد مطور تم إضافته للمرة الأولى للفيجوال بيسك 2005،سعى لتغيير كل ما كان.

الإعدادات في الفيجوال بيسك2008.Settings in Visual Basic 2008

نظام الإعدادات في الفيجوال بيسك2008 متعدد الملفات،معتمد على XML،محدد النوع بقوةstrongly typed ،وتضاهي التراكيب السهلة الإدارة.

تتضمن منهجية ملفها المركز الميزات والفوائد التالية:

.يتم تخزين البيانات في تنسيق XMLمن أجل المعالجة الفعالة بواسطة مكتبات الدوت نت.على الرغم من أنها ليست صورة حرة لنص،فإن XMLغالباً ليست صعبة عند الحاجة إلى التحديث اليدوي الذي يتم عمله من قبل الإنسان.

.البيانات المخزنة في كل ملف إعدادات خاص محددة النوع بقوة،تقلل الأخطاء الناتجة عن معالجة البيانات الغير صحيحة.

.يتم إدارة الإعدادات على كل تطبيق،كل مستخدم وحتى على كل قاعدة إصدار مجمع لتعزيز الأمن وتقليل التعارضات.تستطيع أيضاً تخزين عدة مجموعات من الإعدادات لكل تطبيق حسب الحاجة،مثل مجموعة إعدادات كل مستند يتم فتحه من قبل تطبيقك.(ولن أناقشه في هذا الفصل،ولكن تستطيع البحث عن SettingsKey” ”propertyمن خلال المساعدة على الشبكة لمزيد من المعلومات حول هذه الميزة).

.تتضمن الفيجوال أستوديو أداة إدارة قريبة(محببة)من المستخدم user-friendly management toolتستخدم لتركيب إعدادات ضمن التطبيق.

.لدى الفيجوال بيسك واجهة بسيطة خاصة بها لجعل المستخدم يستخدم ويحدث الإعدادات وقت التنفيذ بشكل أسهل.

ولكن ليس جميعها للمتعة واللعب.كمطور، يقع عليك حمل ثقيل،مثل إظهار أسماء ذات معنى معبرة لكل إعداد(مثل "موقع النموذج الرئيسيMainFormLocation "،"اتصال قاعدة البياناتDatabaseConnection "،إلخ من الأسماء المعبرة)

تظهر الإعدادات الفعلية في ملفات XMLمبعثرة على كامل ملفات النظام:

.وقت التصميم كل الإعدادات التي تعمل على إنشاءها يتم تخزينها في ملف*Settings.settings* المخزن في الدليل الفرعي *My Project* لمجلد كودك المصدري.إليك ملف *Settings.settings*كما هو موجود حتى الآن في مشروع المكتبة:

```
<?xml version='1.0' encoding='utf-8'?>
<SettingsFile xmlns="http://schemas.microsoft.com/VisualStudio/2004/01/settings"
CurrentProfile="(Default)" UseMySettingsClassName="true">
  <Profiles>
    <Profile Name="(Default)" />
  </Profiles>
  <Settings />
</SettingsFile>
```

.وقت التنفيذ، تظهر جميع الإعدادات الخاصة بالمستخدم في ملف *user*.config،يتم تخزينها نموذجياً في:

C :\Documents and Settings\<user>\Local Settings\Application Data\<company>\<appdata>\<version>

حيث*<user>* اسم مستخدم ويندوزWindows username ،*<company>* هو اسم الشركة المسجل في المجمعassembly ، *<appdata>* هو تجمع من القيم يساعد على تميز إعدادات معتمدة على لاستخدام،و*<version>* هو الأرقام الأربع للإصدار الخاص بالمجمع(رقم الإصدار).من الواضح وكأنه مكان صعب لتخزين الإعدادات،ولكن يحفظ الأشياء جميلة ومرتبة.(موقع ملف *user*.config نوعاً ما مختلف إذا ما نشرت تطبيق باستخدام ClickOnce،هذه الطريقة سيتم إنشاء الله شرحها في الفصل 25).

ربما تتسآل فيما إذا كان هذا يشارك في تضخم القرص. الجواب : نعم!فكل مرة توسع(تكبر)رقم إصدار تطبيقك،تعمل الدوت نت على إنشاء ملف إعدادات جديد ليتم شحنه مع هذا الإصدار الجديد.توجد طريقة لتخفيف هذا نوعاً ما،ولكن مع محركات أقراص صلبة(هاردات)120غيغابايت،فلا أحد سيتذمر(يشتكيcomplaining) فيما يخص مساحة القرص المستخدمة بعد الآن.

.بعض الإعدادات تكون متمركزة حول التطبيقapplication-focused، ويتم تقديمها إلى جميع مستخدمي التطبيق على محطة(شبكة عمل)خاصة.ويتم تخزينها في الملف*app.config* الذي يظهر في نفس مجلد مجمعك القابل للتنفيذ assembly's executable. وتظهر الإعدادات في تفرع XML مسمى<applicationSettings> ضمن هذا الملف.الإعدادات المتمركزة حول التطبيق Application-focused لايمكن تعديلها بواسطة التطبيق،عليك تحديث الملف *app.config* يدوياً لإجبار التغيرات.

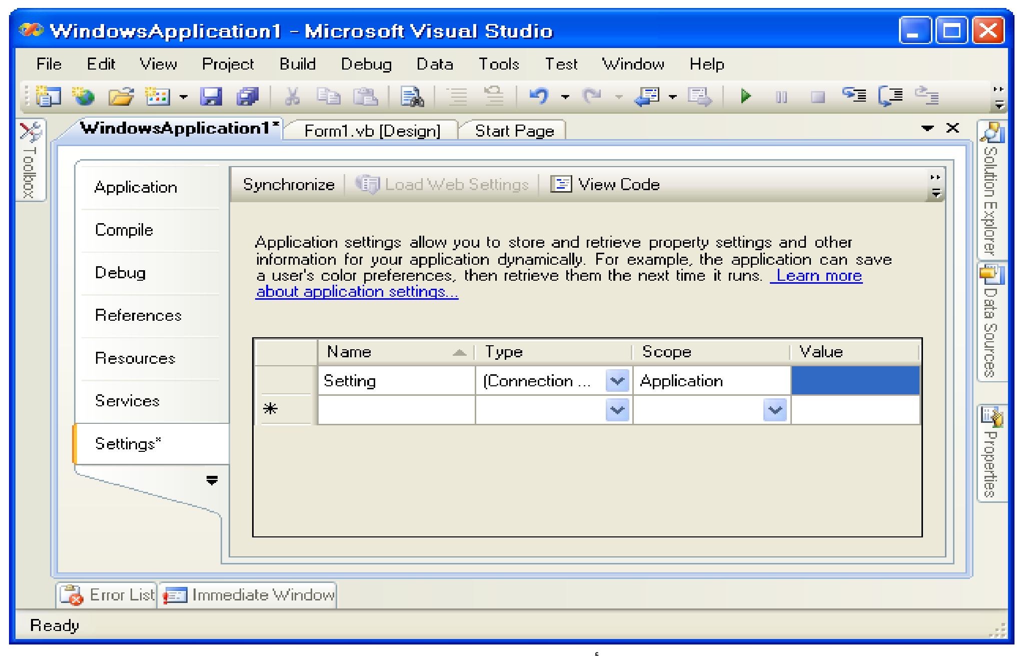
نظام الإعدادات settings system مكان عظيم لتخزين حالة،أشياء تريد من البرنامج أن يتذكرها من المرة الأخيرة التي تم تشغيلها فيه،ولكن يجب أن لايكون هذا صعب التشفير(صعب التكويدhardcoded)في الكود المصدري.

إضافة إعدادات إلى المشروع **Adding Settings to a Project.**

توفر نافذة خصائص المشروع في الفيجوال أستوديو2008 سيطرة مركزية على إعدادات تطبيق ما.لوحة الإعدادات لهذه النافذة مبينة في الشكل التالي،وهي توفر إمكانية الوصول إلى الإعدادات المخصصة للتطبيق.

لإضافة إعدادات،اكتب في اسمهاName ، اختر نموذج نوع Typeبياناتها من القائمة المنسدلة،اختر المجال Scope (مستخدم User أو تطبيقApplication)،وأدخل قيمها Valueباستخدام القيم المتاحة من قبل المحرر من أجل النوع المختار.تتضمن قائمة النوع العديد من الاختيارات الافتراضية،من ضمنها أنواع بيانات الفيجوال بيسك القاعديةthe basic Visual Basic data types، الخطوطfonts، الألوانcolors، والقياسات ذات الصلة بالرسمdrawing-related sizes . ومضمن أيضاً نوع"(نص الاتصالConnection string)" والذي عند اختياره،يمكِّن باني نص خاصيات الاتصال Connection Properties string builder في عمود القيمة Value.

إنه لمن الهام أن تختار النوع الصحيح لكل إعداد مخزنstored setting، وإلا، فإن كمبيوترك سينفجر.عملياً،إني أعتقد أنهم نسقوا هذا في وقت متأخر.وهو كذلك،لأن جميع الإعدادات محددة النوع بقوةstrongly typed. إذا وضعت النوع إلى انتغرInteger، لن تكون قادر على إقحام الكلمة *None*هناك كإشارة خاصة كما يمكنك عمله مع الملفINI. تستطيع اختيار أي نوع دوت نت متاح(محقق) valid .NET type من أجل نوع البياناتdata type، على الرغم من أن الأنواع المعقدة بدون محرراتها الخاصة بها own custom editors، سيتطلب منك وضع قيمها من خلال الكود.



ما الذي يحدث عند إضافة إعداد جديد لمشروع الفيجوال؟لنستكشف ذلك.سأعمل على إضافة إعدادين إلى مشروع نماذج ويندوز:انتقر Integer مسمىWarningLimit، و System.Drawing.Font مسمىNoticeFont (شاهد الشكل التالي).



Name	Type	Scope	Value
WarningLimit	Integer	User	25
NoticeFont	System.Draw...	User	Arial; 14.25pt; style=Bold

كما تعلم مسبقاً،الفيجوال أستوديو هي مجرد إزار قريب من المستخدم حول كود الدوت نت،ولوحة الإعدادات Settings panel لا تختلف عن ذلك.لذلك تحدث التغيرات الحقيقة في مكان ما ضمن الكود،أو بدقة أكثر،في كل من الكود codeوملف *Settings.settings* ذو الصلة.إذا ما "أظهرت جميع الملفاتShow All Files" في مستكشف الحلول،ومدد My Project متبوعاً بـ *Settings.settings*،ستجد أن ملف XMLهذا لديه ملف كود فيجوال بيسك مصدري خاص به *Settings.Designer.vb* .

إذا فتحت الملف*Settings.Designer.vb*، ستجد الكود الجزئي التالي:

```
Namespace My
        <Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(),  _
Global.System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.SettingsDesigner.Settings
SingleFileGenerator", "9.0.0.0"),  _
Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrowsableState.Advanced
)>  _
    Partial Friend NotInheritable Class MySettings
        Inherits Global.System.Configuration.ApplicationSettingsBase
        Private Shared defaultInstance As MySettings =
CType(Global.System.Configuration.ApplicationSettingsBase.Synchronized(New MySettings),MySettings)
    #Region "My.Settings Auto-Save Functionality"
#If _MyType = "WindowsForms" Then
    Private Shared addedHandler As Boolean
    Private Shared addedHandlerLockObject As New Object
    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),
Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrowsableState.Advanced
)> _
    Private Shared Sub AutoSaveSettings(ByVal sender As Global.System.Object, ByVal e As
Global.System.EventArgs)
        If My.Application.SaveMySettingsOnExit Then
            My.Settings.Save()
        End If
    End Sub
#End If
#End Region
         Public Shared ReadOnly Property [Default]() As MySettings
            Get
#If _MyType = "WindowsForms" Then
               If Not addedHandler Then
                    SyncLock addedHandlerLockObject
                        If Not addedHandler Then
                            AddHandler My.Application.Shutdown, AddressOf AutoSaveSettings
                            addedHandler = True
```

```
                        End If
                    End SyncLock
                End If
#End If
                Return defaultInstance
            End Get
        End Property
        <Global.System.Configuration.UserScopedSettingAttribute(),  _
         Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.Configuration.DefaultSettingValueAttribute("25")>  _
        Public Property WarningLimit() As Integer
            Get
                Return CType(Me("WarningLimit"),Integer)
            End Get
            Set
                Me("WarningLimit") = value
            End Set
        End Property
        <Global.System.Configuration.UserScopedSettingAttribute(),  _
         Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.Configuration.DefaultSettingValueAttribute("Arial, 14.25pt, style=Bold")>  _
        Public Property NoticeFont() As Global.System.Drawing.Font
            Get
                Return CType(Me("NoticeFont"),Global.System.Drawing.Font)
            End Get
            Set
                Me("NoticeFont") = value
            End Set
        End Property
    End Class
End Namespace
```

عملت على إخراج الكثير من الكود الزائد.إن المذهل كثرة الكود الذي تحمله ميكروسوفت في المواصفات المكتوبة مقدماً،وليس من الممكن حقاً معرفة ما يجري بالداخل.أستطيع أن أخمن ما تعمله المواصفة DefaultSettingValueAttribute بالنسبة لكل إعداد(تسند القيمة الافتراضية الأولية للإعداد)،ولكن بعض المواصفات الأخرى غامضة.

ولكن باقي الكود واضح تماماً.تولد الفيجوال أستوديو خاصيتين ضمن الفئة My.MySettings، خاصيات مسماة-بشكل مميز وكافي- WarningLimitوNoticeFont .إليك مدخلة الخاصية NoticeFont.

```
Public Property NoticeFont() As Global.System.Drawing.Font
            Get
                Return CType(Me("NoticeFont"),Global.System.Drawing.Font)
            End Get
            Set
                Me("NoticeFont") = value
            End Set
         End Property
```

لن تجد أية أعضاء فئات خاصة تخزن قيم WarningLimit وNoticeFont المخفية.بدل ذلك،في مكان ما أخر في هذه الفئة الخاصة وهو الخاصية الافتراضية (مسماة Item)والتي تمنح gets وتضع setsلك قيمة خاصية محددة defined property value، يمكن الوصول إليها من خلال Me("something").

الإعدادات المتاحة من خلال هذه الخاصية الافتراضية يتم تحميلها مباشرةً من XML مخزن في ملف*Settings.settings*. (هذا الملف يتم ترجمته ضمن التطبيق،ليس عليك توزيع*Settings.settings* مع التطبيق).إليك محتوى من ذلك الملف بالنسبة لقيمتي التركيب الجديدتين الخاصتين بنا:

```xml
<?xml version='1.0' encoding='utf-8'?>
<SettingsFile xmlns="http://schemas.microsoft.com/VisualStudio/2004/01/settings"
CurrentProfile="(Default)" GeneratedClassNamespace="My" GeneratedClassName="MySettings"
UseMySettingsClassName="true">
  <Profiles />
  <Settings>
    <Setting Name="WarningLimit" Type="System.Int32" Scope="User">
      <Value Profile="(Default)">25</Value>
    </Setting>
    <Setting Name="NoticeFont" Type="System.Drawing.Font" Scope="User">
      <Value Profile="(Default)">Arial, 14.25pt, style=Bold</Value>
    </Setting>
  </Settings>
</SettingsFile>
```

كل إعداد يحتوي مواصفات attributesأو مدخلاتentries : اسم Name متميز،نوعType، مجالScope، وقيمة Value، تطابق الأعمدة الأربع التي ظهرت في محرر إعدادات الفيجوال أستوديو.

My.Settings

تنشئ الفيجوال بيسك حالة من فئة My.MySettings ولقد رأيناه منذ قليل في الكود السابق،وتجعلها متاحة كـ My.Settings.فعندما تضيف إعدادات إلى مشروعك،فإنها تصبح أعضاء فئة محددة النوع بقوة لـ My.Settings.للوصول إلى واحد،ببساطة أشر إليه مباشرةً في كودك.

```
MsgBox("The font for notices is: " & My.Settings.NoticeFont.ToString())
```

مخرجات هذا الكود تظهر في الشكل التالي.إن My.Settings.NoticeFontهي حالة حقيقية لـ System.Drawing.font تستطيع استخدامها مثل أي حالة خط أخرى. تستطيع تعديل القيمة لأي إعداد ممتد كـ"مستخدمUser "،وتمنحه قيمة جديدة محجوزة لاستخدامك التالي لتطبيقك(على سبيل المثال الاستخدام التالي للمستخدم الحالي للتطبيق).

```
My.Settings.WarningLimit = 30
```



جميع التغيرات المعمولة لهذه الإعدادات يتم حفظها بشكل آلي إلى ملفات user-specific setting بشكل افتراضي.إذا كنت لاتريد تحديث ما تم حفظه بشكل آلي،ضع My.Application.SaveMySettingsOnExit لأن يشير إلى خطأFalse ،ومن ثم عندما تكون جاهز لحفظ إعدادات جديدة،استخدم الطريقةMy.Settings.Save .

تأتي الإعدادات بثلاث مذاقات لذيذة:افتراضيdefault، المستديمة persisted، وحاليcurrent. الإعدادات الافتراضية Default settings هي تلك القيم المعرفة من قبل المبرمج من خلال محرر إعدادات الفيجوال أستوديو .تتضمن الإعدادات المستديمة Persisted settings التغيرات المحفوظة إلى إعدادات معينة، والإعدادات الافتراضية لتلك التي لن يتم تعديلها من قبل المستخدم.تتضمن الإعدادات الحالية Current settings أي تغيرات يتم عملها للإعدادات خلال الدورة الحالية ولكن لم يتم حفظها بعد. تستطيع التلاعب بهذه الحالات باستخدام أعضاء الكائنMy.Settings :

- الطريقة "حفظ Save، كما ذكرت سابقاً،تحفظ جميع الإعدادات الحالية إلى حالة مستديمة.
- الطريقة إعادة التحميل Reloadتجدد أي قيم حالية بالنسخ المستديمة persisted versions .
- الطريقة "إعادة التنضيد Resetتزيل جميع الإعدادات الحالية والمستديمة وترجع مدخلات تركيب إلى القيم الافتراضية.

واحدة من السمات الأغرب للإعدادات هي أنها نوعية الإصدار.إذا حررت تطبيقك كإصدار 1.0.0.0،ومن ثم فيما بعد حررت الإصدار 1.1.0.0،كل مستخدم سيفقد جميع الإعدادات المستديمة persistedالسابقة.عملياً،لن تفقد،ولكنها ستكون ملتصقة في منطقة1.0.0.0-land.إذا كنت تريد أن تملك دائماً الإعدادات الأحدث up-to-date كلما تم تغيرها من قبل المستخدم،سيكون عليك التأكد من أن الإعدادات الأقدم"تم ترقيتهاupgraded " عند تثبيت نسخة جديدة.يتضمن My.Settings طريقة ترقية Upgrade تقوم بهذا العمل لصالحك.ولكن إذا نصب المستخدم النسخة الأحدث ورقى الإعدادات،وعمل تغييرات على هذه الإعدادات،ومن ثم استدعى upgradeمرةً أخرى،فأي تغيرات معمولة منذ الترقية الأخيرة سيتم فقدها.

لتستوعب(أو تتجنبget around)هذه المشكلة،سيرقي الكود الإعدادات عندما تظهر نسخة جديدة فقط.الطريقة الأسهل لعمل هذا هي بتضمين إعداد يستدعي شيء ما مثل SettingsUpgradedووضعها إلى خطأFalse. اختبر هذه العلامة flagقبل استدعاء Upgrade.فإذا بقيت خطأFalse، فمن الآن استدعاء Upgrade.حالما يرقي الكود الإعدادات،غير SettingsUpgradedإلى صوابTrue .

```
If (My.Settings.SettingsUpgraded = False) Then
        My.Settings.Upgrade()
        My.Setttings.SettingsUpgraded = True
End If
```

هذه الحاجة إلى ترقية الإعدادات في أي وقت حتى ولو تم عمل تغيرات على رقم النسخة minorالمجمع ما تبدو قليلاً أعلى القمة.ولكن من الضروري دعم هدف الدوت نت: التنصيب جنباً إلى جنب.على المستخدم أن يكون قادر على تنصيب نسختين من تطبيق على نفس الكمبيوتر،واستخدام كل واحد دون التداخل من الآخر. إن تخزين إعدادات نوعية الإصدار version-specific settings يساعد على تحقيق هذا الهدف.

الإعدادات المربوطة(المقيدة).Bound Settings

على الرغم من استخدام وتحديث قيم تركيبك الخاصة يمكن أن تكون مثيرة،وحتى ما هو أكثر إثارةً هو أن الحقول في نماذج ويندوز وأدواتها المناسبة يمكنها أن تتفاعل مع الإعدادات المستديمة بشكل آلي.بربط *binding* نموذج form وخاصيات أداة نوعية control specific properties إلى نظام الإعداداتsettings system، فإن الفيجوال بيسك تحفظ وتجدد بشكل آلي أولويات تحكم المستخدم user-controlled preferences ضمن واجهة المستخدم.

استخدام نموذجي لربط الإعدادات هو أن يكون على التطبيق تذكر أين يظهر نموذج خاص على الشاشة عندما تم تشغيل التطبيق في المرة الأخيرة.تحفظ خاصية الفورم Location موقع الفورم على الشاشة.تسجيل هذه القيمة إلى الإعدادات يتطلب خطوتين.أولاً،إنشاء إعداد من نوع System.Drawing.Point لحفظ قيمة الموضع المستديمة.ثانياً،الإشارة في خاصيات الفورم أن قيمة ذلك الموضع سيتم وصلها إلى مدخلة الإعدادات الجديدة.

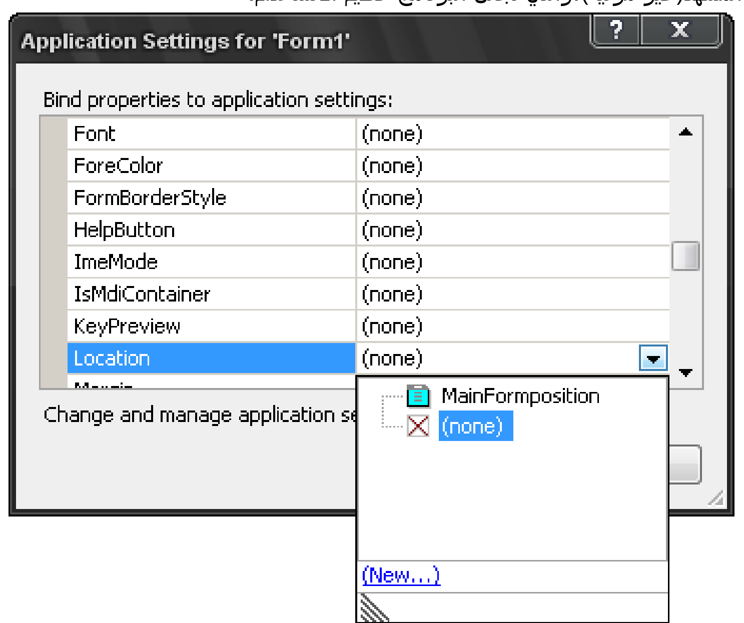
إتمام الخطوة الأولى يتم بإضافة إعداد مستخدم بمدى userونوع System.Drawing.Point في لوحة الإعدادات Settings لخاصيات المشروع.لنسميها MainFormPosition ونترك حقل القيمة Valueفارغ حالياً.

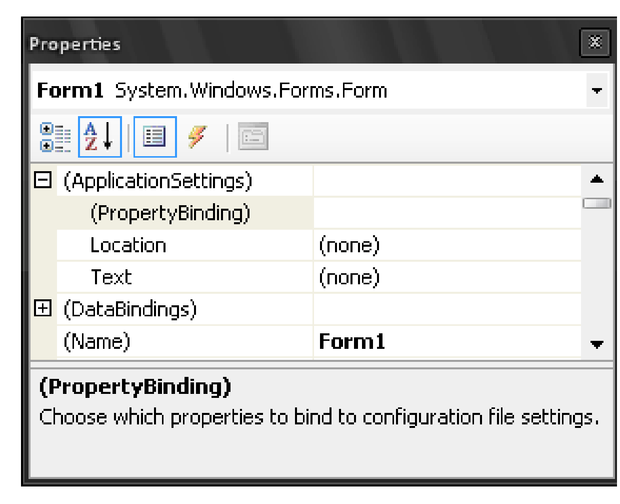
عد إلى محرر الفورم واختر كائن الفورم نفسه،ومن ثم تمكن من الوصول إلى لوحة الخاصيات.مدد خاصية ApplicationSettings لإيجاد الخاصية الجزئية PropertyBinding.بالنقر على زر"..." لهذه المدخلة يعرض حوار إعدادات التطبيق.تظهر معالجة هذا الاختيار في الشكل التالي.

اعمل على إيجاد مدخلة Locationفي القائمة،واختر" MainFormPosition "من أجل قيمتها. حالياً،كل مرة تشغل التطبيق فسيحوي هذا الإعداد المقيد،ستتذكر الفورم المعدلة موضعها السابق.

الخلاصة.

كما مع XML،نظام إعدادات الدوت نت هو واحد من الميزات الداخلية،وخلف المشهد(غير مرئية)،والتي تجعل البرنامج عظيم الاستخدام،





مشروع Project.

بالطبع سنعمل على إضافة إعدادات إلى مشروع المكتبة في هذا الفصل ،ولكن سنعود أيضاً ونستخدم بعض من تلك الإعدادات في الكود والتي أدخلتها مسبقاً كقيم ثابتة. لقد بذلت جهداً كبيراً فيما إذا كنت سأستخدم قيم تركيب(إعداد)بمجال المستخدم user-scoped أو بمجال التطبيق application-scoped من أجل بعض الإعدادات التي تتغير بشكل نادر،مثل نص اتصال قاعدة البيانات.قررت في النهاية استخدام إعدادات المستخدم لذلك يمكن تعديلها من خلال ميزات التطبيق.إن إعدادات مجال التطبيق Application-scoped settings هي للقراءة فقط ويمكن تحديثها من خارج البرنامج فقط،لذلك تم إبعاد تلك الفكرة.المتوقع من إعدادات مجال التطبيق هو أن مدير النظام system administrator سيديرها،إما باستخدام المفكرة على ملفXML ، أو من خلال أداة إدارية ما.بما أننا لن نضيع الوقت في مشروع هذا الكتاب لكتابة أداة إدارة مفصولة،سنحفظ كل شيء على مستوى المستخدم ونتيح التعديل من خلال برنامج المكتبة الرئيسي.

تحديث التوثيق التقني.Update Technical Documentation

لنعمل على توثيق الإعدادات المستخدمة من قبل التطبيق في مجموعة موارد المشروعproject's Resource Kit. أضف المحتوى التالي إلى ملف معالجة نصوص مجموعة الموارد .

إعدادات المستخدم.User settings

يستخدم مشروع المكتبة نظام إعدادات الفيجوال بيسك لتعقب قيم حالة معينة من قبل المستخدم والمحفوظة أثناء استخدام التطبيق.يتم تخزين هذه الإعدادات في ملف في دليل المستخدم *C:\Documents and Settings* (أو ما يكافئه) في تنسيق يملى من قبل الدوت نت.التالي قائمة بالإعدادات المميزة من قبل مشروع المكتبة:

DBConnection (String)

نص اتصال بتنسيق مناسب يحدد قاعدة بيانات سكول سرفر المستخدمة من قبل التطبيق.إذا ما فقد،فإن التطبيق سيسأل عن موقع قاعدة البيانات عند البدء.

HelpFile (String)

يشير إلى UNC(Uniform Naming Convention (أو موقع معتمد على حرف السواقة لملف تعليمات التطبيق الأساسي على الشبكة. مع الامتداد *.chm* .

HelpFileAdmin (String)

يشير إلى UNCأو موقع معتمد على حرف السواقة drive letter-based locationلملف تعليمات التطبيق الإداري على الشبكة.بالامتداد *.chm* .

HideLogin (Boolean)

يشير إلى فيما إذا يجب أن يكون(زر "تسجيل الدخول"في الزاوية العلوية اليسارية للفورم الرئيسية لمشروع المكتبة) غير مرئي عندما يكون العرض في نمط الزبون(العميل).استخدم صواب Trueلإخفاء الزر أو خطأ False لإظهاره.إذا ما فقد هذا الحقل،يتم إسناد خطأFalse .

MainFormPosition (System.Drawing.Point)

موضع نموذج مشروع المكتبة الرئيسي على الزاوية العلوية اليسارية.وهذه القيمة يتم تحديثها كل مرة يتم إغلاق التطبيق.

ReceiptPostlude (String)

أية بيانات حرفية خام(أولية) ليتم إرسالها إلى الطابعة المستقبلة عند الانتهاء من كل بطاقة(تذكرة ticket).يمكن أن يتضمن هذا النص الحروف الخاصة التالية:

\n

حرف سطر جديد(آسكي10: ASCII 10)

\r

حرف عودة المشيرة (آسكي13: ASCII 13)

\e

حرف هروب (آسكي27 : ASCII 27)

\x??

أي قيمة آسكي ،حيث؟؟ شفرة ست عشرية لحرفين.

\\
رمز الشرطة المعكوسة(\ )

ReceiptPrinter (String)

اسم مسار الطابعة المستقبلة المستخدمة من قبل كمبيوتر(أو محطة شبكية محلية)لطباعة وصل استلام مستخرجات عميل patron checkout receipts ووصل المدفوعاتpayment receipts .

ReceiptWidth (Integer)

العرضwidth ،بالحروف،لكل سطر على الطابعة المستلمة.إذا كان فارغ أو مفقود،يستخدم البرنامج العرض الافتراضي 40 حرف.

ReportConfig (String)

يشير إلى UNCأو موضع معتمد على حرف السواقة لملف XMLتركيب تقرير .هذا الملف لديه تنسيق موصوف في مقطع"ملف تركيب تقرير" لهذا المستند.ويشير هذا الملف إلى التقارير المتاحة في التطبيق.

SettingsUpgraded (Boolean)

عند ترقية التطبيق من إصدار أقدم،تشير هذه اليافطة فيما إذا الإعدادات المصاحبة للإصدار الأقدم قد تم ترقيتها ضمن النسخة الجديدة.افتراضاتها خطأ False من أجل جميع الإصدارات.

UseReceipts (Boolean)

تشير فيما إذا الإيصالات المطبوعة يجب أن تستخدم في هذا الكمبيوتر(الشبكة المحلية).إذا ما فقد هذا الحقل،يتم إسناد خطأ False.

هذا الوصف التقني يظهر في مستند مجموعة الموارد التقنيةTechnical Resource Kit documen،بالأصل تم تطويرها في الفصل الرابع وتم تحديثها في الفصول اللاحقة.بعض المحتوى تم إضافته هنا بالإشارة إلى أن المحتوى التقني والميزات لن يتم إضافتها حتى الفصول اللاحقة،لذلك لاتهدر الكثير من الوقت بالتفكير حول الميزات التي قد تظن أنك نسيتها.

إضافة الإعدادات.Add the Settings

بما أننا نعلم جميع الإعدادات التي سنضيفها إلى التطبيق،دعنا نضيفها الآن. افتح نافذة خاصيات المشروع project properties واختر تبويب "إعدادات Settings ."أضف كل إعداد إلى التطبيق باستخدام الجدول التالي.فإذا كان الجدول لا يحتوي قيمة من القيم اترك حقل هذه القيمة فارغة كما هي في محرر الإعدادات.

Nameالاسم	Typeالنوع	Scope المجال	Valueالقيمة
DBConnection	String	User	
HelpFile	String	User	
HelpFileAdmin	String	User	
HideLogin	Boolean	User	False
MainFormPosition	System.Drawing.Point	User	
ReceiptPostlude	String	User	
ReceiptPrinter	String	User	
ReceiptWidth	Integer	User	40
ReportConfig	String	User	
SettingsUpgraded	Boolean	User	False
UseReceipts	Boolean	User	False

تأكد من كتابة اسم الإعداد كما تم جدولته.لن يكون التطبيق قادر على مطابقة الأسماء الغير صحيحة.

موقع الفورم الرئيسي.Positioning the Main Form

لقد بينت لك كيفية وصل قيمة خاصية أداة أو فورم إلى واحدة من الإعدادات سابقاً في هذا الفصل،لذلك لنعمل على عمل هذا عملياً في المشروع.سنعمل على وصل خاصية موقع الفورم الرئيسي إلى إعدادMainFormPosition. لإعادة تنشيط ذاكرتك اتبع الخطوات التالية لتمكين هذا الوصل.

1.افتح الفورم الرئيسي في عرض التصميمin Design view *MainForm.vb* .
2.تأكد من اختيار الفورم نفسها وليس أحد أدواتها التابعة.
3.في لوحة الخاصياتProperties ، مدد الخاصية" ApplicationSettings ."
4.اختر الخاصية الفرعية" PropertyBinding "وانقر على الزر"..."."
5.حدد الخاصية " Location"في قائمة الربط .
6.اختر الإعداد MainFormPositionمن أجل قيمة الخاصية" Location ."فستكون الخيار الوحيد المتاح،بما أنها الوحيدة التي عرفناها من نوع System.Drawing.Point.
7.انقر الزر "موافق OK " لتمكين الربط.

إخفاء واستخدام الإعدادات.Caching and Using Settings

على الرغم من أنه حالما يتم إغلاق الإعدادات يتم كتابة " My.Settings.*something* "في الكود،بعض الإعدادات يمكن أن تكون بشكل أولي غير معرفة،ويمكن أن يورط استخدامها بالكثير من الكود المكرر والذي يختبر صحة الإعدادات.لتقليل الكود ودوران المعالج بشكل عام،سنعمل على إخفاء بعض الإعدادات من أجل الاستخدام السهل على طول التطبيق. دعنا نضيف ثلاث متغيرات عامة لإخفاء بعض الإعدادات.افتح الوحدة البرمجية *General.vb* وأضف هذه المتغيرات الثلاث كأعضاء لهذه الوحدة.

```
Public MainHelpFile As String
Public MainAdminHelpFile As String
Public FineGraceDays As Integer
```

لنمنح هذه المتغيرات قيم أولية في الطريقةInitializeSystem ،حيث يعمل الكود على عمل إسناد أولي لبعض القيم الأخرى.أضف العبارات التالية إلى ذلك الروتين في الوحدة البرمجية General.

```
FineGraceDays = -1
'تحديد ملف المساعدة عبر الشبكة هنا
MainHelpFile = My.Settings.HelpFile & ""
MainAdminHelpFile = My.Settings.HelpFileAdmin & ""
```

في فصل سابق،عملنا على تخزين بعض الإعدادات في الجدول SystemValue والذي يطبق على جميع الكمبيوترات(الشبكات المحلية)التي تتصل بقاعدة البيانات.بما أننا نعمل على إخفاء إعدادات على أية حال،سوف نضيف بعض الكود لإخفاء هذه القيم المخزنة في قاعدة البيانات لذلك ليس علينا الحفاظ على فتح وإغلاق قاعدة البيانات.أضف الطريقة LoadDatabaseSettings إلى الوحدة البرمجية General.

```
Public Sub LoadDatabaseSettings()
        ' الحصول على بعض القيم التي على مستوى النظام من خزانة قاعدة البيانات
        Dim holdText As String
        On Error Resume Next
        ' الحصول على الموقع الافتراضي
        holdText = GetSystemValue("DefaultLocation")
        If (holdText = "") Then holdText = "-1"
        DefaultItemLocation = CInt(holdText)
        ' الحصول على العدد الأعظمي من متطابقات البحث
        holdText = GetSystemValue("SearchLimit")
        If (holdText = "") Then holdText = "-1"
        SearchMatchLimit = CInt(holdText)
        ' الحصول على عدد أيام الانتظار قبل فرض الغرامات
        holdText = GetSystemValue("FineGrace")
        If (holdText = "") Then holdText = "-1"
        FineGraceDays = CInt(holdText)
    End Sub
```

سنعمل على استدعاء هذا الروتين خلال بدء تشغيل التطبيق،تماماً بعد أن نفتح ونؤكد قاعدة البيانات.أضف الكود التالي إلى نهاية معالج الحدث MyApplication_Startup. تذكر أن هذا المعالج موجود في ملفApplicationEvents.vb ،وهو واحد من الملفات المخفية عادة عن العرض في مستكشف الحلول.

```
'تحميل بعض الإعدادات التي تقطن في قاعدة البيانات
            LoadDatabaseSettings()
```

حان الوقت لاستخدام الإعدادات بشكل عملي.يشير الإعداد My.Settings.HideLogin فيما إذا زر"تسجيل الدخول " Login (ActLogin)على الفورم الرئيسية لتطبيق المكتبة سيظهر أم لا عند تشغيل نمط غير المدير(أو غير أمين المكتبة).سيبقى المدير قادر إظهار الزر"تسجيل الدخول "من خلال المفتاح F12،حتى ولو كان الزر مخفي.في بيئة ما حيث يكون المستخدم غير معروف،سيكون النظام بشكل صامت أكثر أمناً إذا ما تم إزالة إغراء temptation زر "تسجيل الدخول".

يتضمن الروتين UpdateDisplayForUserفي فئة الفورم الرئيسية MainFormكود من أجل نمط المستخدم(LoggedInUserID = –1)ومن أجل نمط المدير(LoggedInUserID <> –1).في مقطع نمط المستخدم(المقطع الأول)،بدل هذا السطر :

```
ActLogin.Visible = True
```

بالكود التالي:

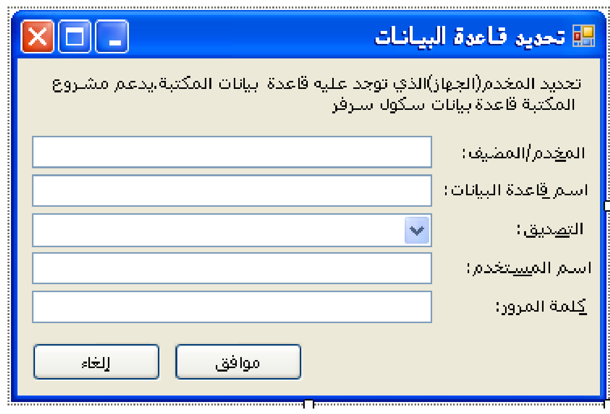
```
'إظهار أو إخفاء زر تسجيل الدخول لكل إعداد
            ActLogin.Visible = Not My.Settings.HideLogin
```

إضافة نموذج التركيب(أو الإعداد). Adding Configuration Forms

حان الوقت لإضافة نماذج تعمل على إدارة جميع إعدادات التطبيق المتنوعة،لكل الإعدادات المخزنة بشكل محلي في ملف الإعدادات المعتمدة على المستخدم،وإعدادات توسيع النظام system-wide settings المخزنة في قاعدة البيانات.معظم الإعدادات بسيطة جداً،مجرد نصوص أساسية،أعداد،وعلامات منطقية،لذلك فلن تربك المدير إذا ما ظهرت جميعها في فورم مفرد. ولكن قبل أن نذهب إلى تلك الفورم،سنعمل على إضافة فورم تتيح لنا إدارة اتصال قاعدة البياناتdatabase connection .

أفكر في استدعاء حوار خاصية الاتصال connection properties dialog التي تستخدمها الفيجوال بيسك لتأسيس نصوص الاتصال.إني واثق من أنها ممكنة،ولكنها توفر طريقة أكثر مرونة من التي نحتاجها في هذا المشروع،فهي تدعم التركيب على قاعدة بيانات غير تابعة لسكول سرفر،والتي هي غير مهمة بالنسبة لمشروع المكتبة(في حال استخدام سكول سرفر أما وقد قمت بتحويل قاعدة البيانات إلى قاعدة بيانات أكسس فيمكن استخدامها).

بالمقابل سنعمل على تصميم نموذج أبسط يعمل على تجميع فقط قيم البيانات التي نحتاجها لبناء نص اتصال قاعدة البيانات.الفورم LocateDatabaseكما يظهر في الشكل التالي.



عملت على إضافة هذه الفورم إلى المشروع(راجع مشروع هذا الفصل).افتح الملفLocateDatabase.vb *LocateDatabase.vb* لرؤية هذه الفورم.أربع من حقول هذه الفورم مجرد حقول لمدخلات نصية أساسية(واحد مع حروف قناع كلمة المرور(بحيث لا تظهر حروف كلمة المرور ويظهر عوضاً عنها رموز مثل رمز النجمة).حقول المدخلات الخمسة،فالتصديق،يتيح للمستخدم الاختيار بين تصديق ميكروسوفت ويندوز Microsoft Windows authentication وتصديق سكول سرفر SQL Server authentication.معظم كود الفورم متوافق لما قد رأيناه في العديد من النماذج الأخرى في السابق،واصل العمل الآن وأضف جميع الكود إلى الفورم الآن.
يحدث العمل الهام في هذه الفورم في حدث التحميل Loadعندما يتم ترجمة نص الاتصال الموجود ضمن حقول إدخال البيانات لمختلفة،وفي الروتين PromptUserحيث يتم إعادة وضع الأجزاء مع بعضها البعض.
توجد عدة طرق مختلفة تمكن من قطع(تقسيم) نص الاتصال ضمن الأجزاء القاعدية.لقد أخذت القاعدة الأساسية فرق تسد divide-and-conquer، مستخرجاً كل مكون مفصول بإشارة مساواة وفاصلة منقوطة. راجع مقطع معالج حدث تحميل هذه الفورم والتي تعمل قطع(تقسيم)واستخراج:

```
Private Sub LocateDatabase_Load(…)…
```

على الرغم من أن الفورم LocateDatabaseتقوم بجميع الترجمة والبناء لنص الاتصال،فهي لا تعمل عملياً على تحديث الإعدادات المحفوظة.بالمقابل،تعمل على إرجاع نص اتصال تم بناءه حديثاً،وتعتمد على الكود المستدعى لحفظه.
الآن،عد إلى محرر تركيب نموذج مفرد،الفورم.*Maintenance.vb*، فهذا الفورم يعمل جميع التعديل على القيم في كل من قاعدة البيانات وبنود My.Settings المحلية.يبين الشكل التالي لوحتين رئيسيتين على هذا النموذج.الإعدادات المركزة المخزنة في قاعدة البيانات هي"توسيع النظام system-wide "والقيم" خاصة بالجهاز أو محطة العمل workstation-specific "هي تلك التي يمكن الوصول إليها من خلال My.Settings.



يبدأ هذا النموذج عمله في معالج حدث التحميلLoad، وهذا الروتين يثبت الخيارات في بعض الحقول المنسدلة،متضمناً قائمة الخطوط. يدور الكود خلال تجميع الخطوط المنصبة جاعلاً إياها متاحة من خلال كائن الـ GDI: System.Drawing.Text.InstalledFontCollection.

```
Dim allFonts As New System.Drawing.Text.InstalledFontCollection
RecordFontName.Items.Add(New ListItemData("<Not Selected>", -1))
        For counter = 0 To allFonts.Families.Length - 1
            RecordFontName.Items.Add(New ListItemData(allFonts.Families(counter).Name, counter))
        Next counter
```

يتضمن الروتين أيضاً كود مشابه لتحميل قائمة الطابعات المثبتة.

```
For Each installedPrinter As String In PrinterSettings.InstalledPrinters
            RecordPrinterLocation.Items.Add(installedPrinter)
        Next installedPrinter
```

حالما يتم تثبيت كل شيء،يعمل الإجراء PopulateCurrentValuesعلى إكمال التمهيد .فيعمل كوده على استخلاص جميع القيم الحالية من قاعدة البيانات ومن الكائن My.Settings.ويخزن هذه القيم في حقول إدخال البيانات المتنوعة على الشاشة. لقد عملت على إضافة الكود الخاص بقاعدة البيانات.تابع واعمل على إضافة الكود الخاص بالإعدادات.

```
LibraryConnection = My.Settings.DBConnection & ""
        RecordDBLocation.Text = GetDBDisplayText(LibraryConnection)
        RecordConfigLocation.Text = My.Settings.ReportConfig & ""
        RecordBasicHelp.Text = My.Settings.HelpFile & ""
        RecordAdminHelp.Text = My.Settings.HelpFileAdmin & ""
        EnableReceipts.Checked = My.Settings.UseReceipts
        RecordPrinterLocation.Text = My.Settings.ReceiptPrinter & ""
        RecordPrinterWidth.Text = CStr(My.Settings.ReceiptWidth)
        RecordPostlude.Text = My.Settings.ReceiptPostlude & ""
        HideLogin.Checked = My.Settings.HideLogin
```



معظم الكود في هذه الفورم يتعامل مع تفاعل المستخدم العادي بينما يكون الفورم قيد الاستخدام.على سبيل المثال،معالج حدث الزر ActDBLocation_Click يعرض نموذج"موقع قاعدة البيانات LocateDatabase " الذي أضفناه سابقاً.أضف الكود المصدري المناسب إلى معالج الحدث هذا.

```
Private Sub ActDBLocation_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ActDBLocation.Click
        ' ----- Prompt for the database connection details.
        Dim newConnection As String
        ' ----- Prompt the user for the new setting.
        newConnection = LocateDatabase.PromptUser()
        If (newConnection = "") Then Return
        ' ----- Store the new value.
        LibraryConnection = newConnection
        RecordDBLocation.Text = GetDBDisplayText(LibraryConnection)
    End Sub
```

تعين العديد من الإعدادات موقع الملفات المستخدمة من قبل التطبيق،مثل ملفات المساعدة عبر النت،يمكن للمستخدم أن يكتب في المسار دليل الملف،أو يستخدم حوار فتح ملف لتحديد الملف بشكل مرئي.لعرض الحوار.عملت على إضافة أداة "حوار فتح ملف OpenFileDialog "مسماة LocateFile.استخدامها هو موضوع إعدادات الخاصيات تعين ملف المتنوعة واستدعاء الطريقة ShowDialog.(راجع الكود المضمن في معالج حدث ActBasicHelp_Click المستخدم لإيجاد ملف المساعدة الغير إداري عبر الشبكة والذي يبدأ بالكود التالي☺

```
Private Sub ActBasicHelp_Click(…) …
```

حالما يكون المستخدم قد عمل الإعدادات تغيرات على الإعدادات المتنوعة،يؤدي النقر على الزر"موافق" إلى حفظ كل إعداد جديد لمنطقة حفظه.عملت على تضمين كود الحفظ المعتمد على قاعدة البيانات في الروتين SaveFormData. وسأتيح لك إضافة كود معتمد على الإعدادات ،قرب نهاية الروتين.(راجع هذا الكود الذي بدايته☺

```
Private Function SaveFormData() As Boolean
```

على الرغم من أن الفورم Maintenance "صيانة"توفر واجهة قريبة من المستخدم للإعدادات المخزنة في قاعدة البيانات،من المحتمل أن تتذكر أننا كتبنا سابقاً كود تحديث سجلات جدول"قيم النظام SystemValue " من خلال الملف*SystemValue.vb* .في الفصل 12،لقد وصلنا تلك الفورم إلى الفورم الرئيسية،ولكن نحن الآن ذاهبون لتغير ذلك المنطق.أولاً،سنعمل على إضافة استدعاء إلى نموذج"قيم النظام SystemValue " من "نموذج الصيانة" في معالج الحدث ActAllValues_Click.

```
Private Sub ActAllValues_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ActAllValues.Click
        ' ----- Let the user edit the list of system values.
        Dim RecordsForm As LIBRARY.ListEditRecords
        ' ----- Edit the records.
        RecordsForm = New LIBRARY.ListEditRecords
        RecordsForm.ManageRecords(New LIBRARY.SystemValue)
        RecordsForm = Nothing
```

```
        ' ----- Refresh the display elements.
        PopulateCurrentValues()
    End Sub
```

ومن ثم سنعود ونغير معالج الحدث AdminLinkValues_LinkClicked في الكود المصدري للفورم الرئيسية MainForm.*vb* . حالياً، يستدعي محرر SystemValue مباشرةً.غيّر ذلك الجزء من كود معالج الحدث بالكود الذي يستدعي النموذج "صيانة" بدلاً عنه.

```
' ----- Access the maintenance portion of the program.
        Maintenance.ShowDialog()
```

الاتصال بقاعدة البيانات المركبة.Connecting to the Configured Database

التغير الأخير في هذا الفصل يستخدم نص الاتصال المركب لتأسيس الاتصال إلى قاعدة البيانات.عندما كتبنا روتين ConnectDatabaseبالأصل في الوحدة البرمجية General،عملنا على إضافة نص اتصال ثابت من أجل جعل البرنامج يعمل فقط.

```
' بناء نص الاتصال، ثابت في الوقت الحالي
connectionString = "Data Source=MHM\SQLEXPRESS;" & "Initial Catalog=C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\library.mdf;Integrated Security=true"
```

الآن متاح لدينا نص اتصال مركب من قبل المستخدم،سنعمل على استخدامه بدل المذكور في الأعلى.التغيرات التي يجب أن نعملها لهذا الروتين نوعاً ما واسعة،لذلك قم فقط بتبديل محتوى الدالة بالكود التالي:

```
Public Function ConnectDatabase() As Boolean
' الاتصال بقاعدة البيانات بصواب العودة في حال النجاح
        Dim connectionString As String
        Dim configChanged As Boolean
        ' التمهيد
        HoldTransaction = Nothing
        configChanged = False
' الحصول على نص الاتصال
        If (Trim(My.Settings.DBConnection & "") = "") Then
' إعلام المستخدم حول الحاجة إلى تركيب قاعدة البيانات
            If (MsgBox("هذه النسخة من التطبيق لم يتم تهيئتها للإتصال إلى " & "قاعدة البيانات, إذا كنت تعرف إعدادات قاعدة البيانات " & "تستطيع تركيبها الآن,هل تريد المواصلة؟", MsgBoxStyle.YesNo Or MsgBoxStyle.Question, ProgramTitle) <> MsgBoxResult.Yes) Then Return False
' الطلب من المستخدم تفاصيل اتصال جديدة
            connectionString = LocateDatabase.PromptUser()
            If (connectionString = "") Then Return False
            configChanged = True
        Else
            connectionString = My.Settings.DBConnection
        End If
TryConnectingAgain:
' محاولة فتح قاعدة البيانات
        Try
            LibraryDB = New SqlClient.SqlConnection(connectionString)
            LibraryDB.Open()
        Catch ex As Exception
' بعض إخفقات قاعدة البيانات
            GeneralError("ConnectDatabase", ex)
' من المحتمل أنها قضايا التركيب فقط
            If (MsgBox("الاتصال إلى قاعدة البيانات يمكن أن يكون قد أخفق بسبب " & "إعدادات التهيئة الغير صحيحة,هل تريد تغير " & "إعداد قاعدة البيانات الحالي؟", MsgBoxStyle.YesNo Or MsgBoxStyle.Question, ProgramTitle) _
              <> MsgBoxResult.Yes) Then Return False
' طلب تفاصيل جديدة
            connectionString = LocateDatabase.PromptUser()
            If (connectionString = "") Then Return False
            configChanged = True
            GoTo TryConnectingAgain
        End Try
' حفظ الإعدادات المحدثة عند الحاجة
        If (configChanged = True) Then _
            My.Settings.DBConnection = connectionString
' في حال النجاح
        Return True
    End Function
```

إذا استخدمت مخدم أكسس"`Provider=Microsoft.Jet.OLEDB.4.0`" نحتاج إلى عمل التغيرات في الروتين `PromptUser` في كود الفورم"تحديد قاعدة البيانات LocateDatabase"،قم بجعل السطر التالي تعليق:

```
' newConnection &= ";Integrated Security=false"
```

واجعل أيضاً السطر التالي تعليق:

```
'newConnection &= ";User ID=" & Trim(RecordUser.Text) &  ";Password=" & Trim(RecordPassword.Text)
```

واعمل التغيرات التالية في السطر التالي:

```
newConnection = "Provider=" & Trim(RecordServer.Text) & ";Data Source=" & Trim(RecordDatabase.Text)
```

والآن اكتب الموفر `Microsoft.Jet.OLEDB.4.0`،في حقل المخدم/المضيف ،اكتب مسار قاعدة البيانات في حقل "اسم قاعدة البيانات" على سبيل المثال توجد قاعدة البيانات عندي على المسار التالي والذي سأكتبه في حقل "اسم قاعدة البيانات":C:\Documents and Settings\Administrator\library.mdb .
ومن ثم انقر موافق.

الأمر gistالأساسي في الكود هو أن الإعداد يتضمن المتغير connectionString لنص الاتصال المصممpersisted ،ويستخدمه لفتح الكائن LibraryDB.يحصل الكود الجديد على نص الاتصال من My.Settings.DBConnection . إذا ومن أجل أي سبب تم فقد نص الاتصال أو فشل في توليد فتح اتصال لقاعدة البيانات،يتم الطلب من المستخدم توفير نص اتصال صحيح من خلال الفورم LocateDatabase.

يعود البرنامج إلى الحالة بحيث تستطيع تشغيله مرةً أخرى.في المرة الأولى التي تشغل فيها التطبيق،فإنه سيطلب من المستخدم تزويد معلومات الاتصال بقاعدة البيانات، القيم التي توفرها يجب أن تطابق الحالة الثابتة الموجودة في الروتين ConnectDatabase.

إذا كنا نستخدم موفر سكول سرفر فيجب أن يكون التالي(لكن لاتنس أنني في المشروع استخدم مخدم أكسس فلا تختلط عليك الأمور):

.ضع المخدم/المضيف Server/Host إلى " MYSERVER\SQLEXPRESS " أو إلى اسم مضيف سكول سرفر الفعلي الخاص بك.

.ضع اسم قاعدة البيانات إلى " Library " أو أي اسم قمت بإسناده سابقاً كاسم لقاعدة بيانات المكتبة.

.ضع التصديق Authenticationإلى " Microsoft Windows "إذا كنت تستخدم أمن ويندوز المتكاملWindows integrated security. إذا كنت بحاجة إلى استخدام نظام أمن سكول سرفر،ضع هذا الحقل إليه أي " SQL Server "،وأدخل اسم مستخدم صحيح ID وكلمة مرور صحيحةpassword .

سنركز في الفصل القادم على تقنيات معالجة الملفات.على الرغم من أنني عملت على تحديث ملف الإعدادات في هذا الفصل،فقد تم عمله بشكل غير مباشر من خلال الميزات الموفرة من قبل إطار عمل الدوت نت.سيناقش الفصل15 قواعد مباشرة أكثر خاصة بمعالجة الملفات.

الملفات والأدلة Files and Directories

تطور البرمجيات في بداية القرن الحادي والعشرون حول المبرمجين حقاً إلى مجموعة من الحالمين(لا أقصد التلاعب بالألفاظ).في الأيام القديمة للكمبيوتر،كان على المطورين ربط البرامج إلى الكمبيوتر يدوياً.فيمكن لحسابات معقدة أن تأخذ أيام لإعدادها.كانت المعاناة حقيقية،والقضايا الأقدم للإلكترونيات الشائعة يتم غربلتها بالمواضيع من قبل المبرمجين السابقين والذين كانوا مولعين في محاولتهم لصناعة خوارزمية حساب أكثر دافعية ومنهجية.

تطورت الحياة بشكل هائل بالنسبة للمبرمجين عندما اقترح جون فون نيومان وآخرين أن باستطاعة الكمبيوتر تخزين المنطق للخوارزميات بشكل داخلي،ومعالجتها مباشرة من الذاكرة بدلاً من التركيبات الصعبة المثيرة في كل مكان.عمل المبرمجين مباشرة على وضع برامجهم ضمن مجموعة كروت وأشرطة ورقية.حتى جاء اختراع الهارد ديسك(القرص الصلب) والتقنيات المتعلقة به.

وهكذا كانت ولادة نظام الملفاتfilesystem، المخزن التركيبي للبرامج والمعلومات على سطح الديسك(القرص). Filesystems أنظمة الملفات قد أصبحت جزء من تقنيات ميكروسوفت منذ تودد بيل غيتس للمرة الأولى ل أي بي إمIBM. لم يكن صدفة أن يرمز الدوس DOS في ميكروسوفت دوس MS-DOS إلى قرص نظام تشغيل *Disk Operating System*. عرف بل كيف كانت أنظمة الملفات الأساسية،وكذلك أنت.في هذا الفصل ،سوف نتكلم حول التفاعل مع الملفات والأدلة،والوحدات الرئيسية التخزينية والتنظيمات في نظام ملفات ويندوزWindows filesystem. وسنرى أيضاً أن بعض تقنيات وميزات الدوت نت توفر معالجة للملفات ومحتوياتها.

إدارة ملف الفيجوال بيسك التقليدي Traditional Visual Basic File Management

عملت الفيجوال بيسك على تضمين ميزات إدارة الملف الهامة منذ الإصدار الأول.في الحقيقة،توجد ميزات في الفيجوال البيسك تتعامل مع معالجة ملف fileودليل directory أكثر نسبياً من أي شيء أخر.معظم الدوال التي تسمح لك بقراءة وتعديل محتوى ملف تستخدم معالجة ملف*file handle*، ومعرفات عددية numeric identifier تشير إلى ملف مفتوح نوعي.ومعالج الملف file handle هذا يتم توليده بالدالة FreeFile، ويجب أن تسود قبل استدعاء أي ميزات ملف فيجوال بيسك تقليدي.

```
Dim fileID As Integer
        fileID = FreeFile()
        FileOpen(fileID, "C:\TestData.txt", OpenMode.Append)
        PrintLine(fileID, "Important output to file.")
        FileClose(fileID)
```

تعمل معالجة ملف معتمدة على معالج ملف بشكل جيد،ولكنها قديمة جداً،وحقيقةً ليست تقنية دوت نت،وليست معتمدة على كائن على الإطلاق(مالم تعتبر أن الإنتغر Integerكائن). وبالتالي فلن أغطيها في هذا الكتاب،أو استخدمها في مشروع المكتبة.يحدد الجدول التالي ميزات الفيجوال بيسك الرئيسية والتي تستخدم معالجات ملف.إذا كنت تريد معرفة المزيد حول الميزات المعتمدة على المعالجة في الفيجوال بيسك،أو إذا كان عملك يتضمن ترحيل تطبيقات فيجوال بيسك سابقة إلى الدوت نت،استخدم هذا الجدول لمساعدتك في إيجاد تفاصيل الميزات الكاملة في المستندات التقنية المزودة مع الفيجوال بيسك.

Visual Basic features that use file handles

Feature	Description
EOF	Returns a Boolean indicating whether the current position in the file is at or past the end of the file. Use this function to determine when to stop reading existing data from a file.
FileAttr	Accesses the file attributes currently set on an open file handle.
FileClose	Closes a specific file opened using a file handle.
FileGet	Retrieves structured data from a file and stores it in a matching object.
FileGetObject	Same as FileGet, but with slightly different data typing support.
FileOpen	Opens a file for input or output.
FilePut	Writes an object to a file in a structured manner.
FilePutObject	Same as FilePut, but with slightly different data typing support.
FileWidth	Sets the default line width for formatted text output files.
FreeFile	Returns the next available file handle.
Input	Retrieves a value previously written to a file using Write or WriteLine.
InputString	Retrieves a specific number of characters from an input file.
LineInput	Returns a complete line of input from a file.
Loc	Returns the current byte or record location in the file.

Lock	Locks a file or specific records in a file so that others cannot make changes.
LOF	Returns the length of an open file, in bytes.
Print	Sends text output to a file.
PrintLine	Sends text output to a file, ending it with a line terminator.
Reset	Closes all files currently opened with file handles.
Seek	Gets or sets the current position in a file.
SPC	This function helps format text for output to columnar text files.
TAB	This function helps format text for output to columnar text files.
Unlock	Removes locks previously set with Lock.
Write	Writes data to a file using a consistent format that can be easily read later.
WriteLine	Same as Write, but ends the output with a line terminator.

معالجة الملفات من خلال التجميعات Manipulating Files Through Streams

يتضمن إطار عمل الدوت نت مقاربة موجهة بالكائنات لقراءة وكتابة الملفات:التجميعات*streams*. كائن التجميع Streamالمجرد موجود في System.IO.Stream،ويعرف وسيط interfaceشامل لقطعة كبيرة من البيانات.ولا يهم مكان البيانات:في ملف،أو في مقطع من الذاكرة،أو في متغير نصي String variable إذا كان لديك مقطع من البيانات يمكنك من قراءة أو كتابة بايت واحد في كل مرة،فتستطيع تصميم فئة تجميع مشتقة derived stream class للتفاعل معه.

ميزات التجميع "Stream". Stream Features

تتضمن الميزات الأساسية لكائن ستريم Streamطرق قراءة Readوكتابة Write والتي تتيح لك قراءة أو كتابة وحدات تخزينية bytes.بما أن البيانات هي قراءة من أو كتابة إلى تجميع ستريمStream، فإن كائن الستريم يحفظ "الموضوع الحالي current position" ضمن الستريم والذي تستطيع تعديله باستخدام الطريقة "بحث Seek"،أو التفحص باستخدام الخاصية "موضع Position". تشير خاصية "الطول Length" إلى حجم البيانات القابلة للقراءة.تعرض الفئة أيضاً انحرافات عن هذه الميزات القاعدية لإتاحة المرونة قدر الإمكان.

ليس كل ستريم يوفر كل الميزات.بعض الستريمات للقراءة فقطread-only ، وبعضها للأمام فقط forward-onlyالتركيبات التي لاتدعم الكتابة writingأو البحثseeking . تجميعات أخرى تدعم جميع الميزات الممكنة.الميزات المتاحة لك تعتمد على نوع الستريم(التجميع) الذي تستخدمه.بما أن الستريم نفسه مجرد،فيتوجب عليك إنشاء حالة من فئاته المشتقة.تعرف الدوت نت عدة ستريمات مفيدة جاهزة للاستخدام:

تجميع ملف .FileStream

كائن FileStreamيتيح لك الوصول إلى محتوى ملف باستخدام الطرق القاعدية لفئة ستريم الشاملةgeneric Stream class. وهي كائنات تجميع ملف تدعم القراءة reading،الكتابةwriting، والبحثseeking، على الرغم من ذلك إذا فتحت ملف للقراءة فقطread-only file، فلن تكون قادر على الكتابة له.

تجميع الذاكرة .MemoryStream

تجميع معتمد على مقطع من صفوف الذاكرة.تستطيع إنشاء تجميع ذاكرة بأي حجم،وتستخدمه بشكل مؤقت لتخزين واستخراج أي بيانات.

تجميع الشبكة.NetworkStream

تجرد هذه الفئة البيانات القادمة من مقبس socket شبكة ما.بينما معظم فئات ستريم المشتقة تقع فيSystem.IO، فإن هذه الفئة تقع في System.Net.Sockets.BufferedStream. تضيف دعم تخزين انتقالي buffering لستريم لتحسين الأداء على الستريمات مع قضايا الكمون(مشاكل الخفاء).إنك تطوق الكائن BufferedStreamحول ستريم أخر لاستخدامه.

التشفير إلى تجميع CryptoStream

يسمح لك هذا الستريم إرفاق موفر خدمة تشفير،ينتج عنها مخرجات مشفرة من مدخلات بسيطة،أو العكس بالعكس.يتضمن الفصل 11 أمثلة تستخدم هذا النوع من الستريم.

DeflateStream *and* GZipStream

يتيح لك استخدام ستريم لضغط أو فك ضغط بيانات عندما يتم معالجتها،وكلها تستخدم خوارزميات ضغط قياسيةstandard compression algorithms .

الستريمات(التجميعات)مفيدة بحد ذاتها،ولكن تستطيع أيضاً ضم تجميعات(ستريمات) بحيث يمكن أن يتم بشكل مباشر تشفير ستريم شبكة،ضغطه وتخزينه في مقطع تجميع(ستريم) ذاكرة.

استخدام ستريم. Using a Stream

استخدام ستريم بسيط،أولاً عليك إنشاءه،ومن ثم تبدأ بقراءة وكتابة البايتات bytesلليسار واليمين.إليك كود بسيط ينقل البيانات إلى ومن ستريم ذاكرة.وهو معتمد على كود ستجده في مستندات ميكروسوفت MSDN لفئة MemoryStream.

```
' ----- The Stream, or There and Back Again.
      Dim position As Integer
      Dim memStream As IO.MemoryStream
```

```
        Dim sourceChars() As Byte
        Dim destBytes() As Byte
        Dim destChars() As Char
        Dim asUnicode As New System.Text.UnicodeEncoding()
        ' ----- Create a memory stream with room for 100 bytes.
'أنشاء تجميع ذاكرةبمجال 100 بايت.
        memStream = New IO.MemoryStream(100)
        ' ----- Convert the text data to a byte array.
'تحويل البيانات النصية إلى مصفوفة بايتات.
        sourceChars = asUnicode.GetBytes("This is a test of the emergency programming system.")
        Try
            ' ----- Store the byte-converted data in the stream.
'تخزين البيانات المحولة للبايت في الستريم.
            memStream.Write(sourceChars, 0, sourceChars.Length)
            ' ----- The position is at the end of the written data.
'الموقع عند نهاية البيانات المكتوبة.
            ' To read it back, we must move the pointer to
            ' the start again.
'لقراءته مرةً أخرى،علينا نقل المؤشر إلى البداية مرة أخرى.
            memStream.Seek(0, IO.SeekOrigin.Begin)
            ' ----- Read a chunk of the text/bytes at once.
'قراءة قطعة من نص/بايتات حالاً.
            destBytes = New Byte(CInt(memStream.Length)) {}
            position = memStream.Read(destBytes, 0, 25)
            ' ----- Get the remaining data one byte at a time,
            ' just for fun.
'الحصول على البيانات المتبقية بايت في كل مرة،فقط من أجل المتعة.
            While (position < memStream.Length)
                destBytes(position) = CByte(memStream.ReadByte())
                position += 1
            End While
            ' ----- Convert the byte array back to a set of characters.
'تحويل مصفوفة بايت مرةً أخرى إلى مجموعة حروف.
            destChars = New Char(asUnicode.GetCharCount(destBytes, 0, position)) {}
            asUnicode.GetDecoder().GetChars(destBytes, 0,position, destChars, 0)
            ' ----- Prove that the text is back.
'إثبات أن النص يتم إرجاعه.
            MsgBox(destChars)
        Finally
            memStream.Close()
        End Try
```

على أمل أن تكون التعليقات قد جعلت الكود واضح.بعد إنشاء تجميع ذاكرةmemory stream ،وضعت مقطع من النص فيه،ومن ثم قراءته باسترجاع(بطريقة تراجعية).(يبقى النص في الستريم،وقراءته لا تزيله).عملياً،كود الستريم سهل جداً.معظم الكود يتعامل مع التحويل بين البايتات والحروف.وإذا بدا أنه ذو محتوى زائد فهو كذلك.

ما خلف بايتات الستريم Beyond Stream Bytes

بالنسبة لي،كل تلك التحويلات بين البايتات والحروف لا تستحق الاهتمام.فعندما أكتب تطبيقات عمل،فإنني أتعامل بشكل نموذجي مع التواريخ،الأعداد والنصوص:أسماء الزبائن، تاريخ الطلبية،كمية المدفوعات،وهكذا.نادراً ما يكون علي العمل على مستوى البايت.على الرغم من أنك تستطيع معالجة الستريمات مباشرةً إذا كنت تريد أو تحتاج حقاً لذلك، ففضاء الأسماء System.IOيتضمن أيضاً العديد من الفئات والتي توفر منطقة فاصلة(انتقالية)أكثر قرباً للمستخدم بينك وبين الستريم.يتم تنفيذ هذه الفئات كقارئات readersوكاتبات writersمميزة لبيانات ستريم الذي يوفر طرق مبسطة لتخزين أنواع بيانات معينة،واسترجاعها مرةً أخرى.يتم تصميم القارئات من أجل اتجاه واحد معالجة البيانات تتم من البداية إلى النهاية.بعد إنشاء أو إمكانية الوصول إلى الستريم،فإنك تغلف ذلك الستريم إما بقارئ أو كاتب،وتبدأ اجتياز نطاق الستريم من البداية.عليك دائما الوصول إلى الستريم الضمني إذا كنت تريد تحكم ذو قابلية أكبر للتعديل عند أي نقطة.

يوجد ثلاث أزواج رئيسية من القارئات والكاتبات:

كاتبات وقارئات ثنائية BinaryReader *and* BinaryWriter

تجعل هذه الفئات من السهل كتابة وفيما بعد قراءة أنواع بيانات الفيجوال بيسك الأساسية إلى ومن ستريم غير نصي(بشكل عام).تتضمن الطريقة BinaryWriter.Write إعادة تعريف لكتابة بايتات،حروف وانتغر بإشارة أو بدون إشارة وبأحجام متنوعة،البولين(المنطقي)،العشري،النصوص،والمصفوفات ومقاطع بايتات وحروف.ومن الغريب عدم وجود إعادة تعريف overloadلقيم التاريخ Date.

يتضمن النظير BinaryReaderالطريقة المنفصلة Readمن أجل كل نوع بيانات قابل للكتابة.ترجع الطريقة ReadDoubleقيمة Double من الستريم،وتوجد طرق مشابهة من أجل أنواع البيانات الأخرى.

كاتبات وقارئات جدولية(تجميعية)StreamReader *and* StreamWriter

تستخدم هذه الفئات بشكل نموذجي لمعالجة ملفات نصية معتمدة على سطر .تتضمن الفئة StreamReader الطريقة ReadLineوالتي ترجع بالسطر التالي في النص في الستريم القادم كسلسلة نصية String قياسية. تتضمن الطريقة StreamWriter.Write ذات الصلة كل إعادات التعريف overloadsلـ BinaryWriter.Write،ولديها أيضاً نسخة(إصدار) يتيح لك تنسيق نص من أجل المخرجات.يتضمن القارئ readerميزات تتيح لك قراءة بيانات حرف حرف، مقطع مقطع،أو كامل الملف في نفس الوقت.

قارئ وكاتب السلاسل النصية StringReader *and* StringWriter.

يوفر هذا الزوج من الفئات نفس الميزات التي يوفرها الزوج StreamReader وStreamWriter،ولكن يستخدم حالة سلسلة حرفية Stringقياسية من أجل تخزين البيانات بدلاً من ملف.

يوفر زوج إضافي TextReader وTextWriter فئة قاعدية من أجل القارئات والكاتبات الغير ثنائية الأخرى.تستطيع إنشاء حالات منها بشكل مباشر،ولكنها تتيح لك معاملة حالات السلسلة الحرفية والستريم للقارئات والكاتبات بشكل عام.بهذه الأدوات الجديدة ،من الأسهل معالجة البيانات غير البايت من خلال الستريمات.إليك إعادة كتابة كود ستريم الذاكرة والذي كتبته سابقاً،والمعدل من أجل استخدام قارئ وكاتب جدولي StreamReader and StreamWriter .

```
' ----- The Stream, or There and Back Again.
        Dim memStream As IO.MemoryStream
        Dim forWriting As IO.StreamWriter
        Dim forReading As IO.StreamReader
        Dim finalMessage As String
        Dim asUnicode As New System.Text.UnicodeEncoding()
        ' ----- Create a memory stream with room for 100 bytes.
        memStream = New IO.MemoryStream(100)
        Try
            ' ----- Wrap the stream with a writer.
            forWriting = New IO.StreamWriter(memStream, asUnicode)
            ' ----- Store the original data in the stream.
            forWriting.WriteLine("This is a test of the emergency programming system.")
            forWriting.Flush()
            ' ----- The position is at the end of the written data.
            ' To read it back, we must move the pointer to
            ' the start again.
            memStream.Seek(0, IO.SeekOrigin.Begin)
            ' ----- Create a reader to get the data back again.
            forReading = New IO.StreamReader(memStream, asUnicode)
            ' ----- Get the original string.
            finalMessage = forReading.ReadToEnd()
            ' ----- Prove that the text is back.
            MsgBox(finalMessage)
        Finally
            memStream.Close()
        End Try
```

من المؤكد أن هذا الكود أفضل بكثير من جميع التحويلات في الكود السابق والذي يجعل العمل في حالة فوضى.(نستطيع حتى تبسيطه أكثر وذلك بالتخلص من جميع أشياء تشفير يونيكود الاختيارية).بالطبع كل شيء مايزال قيد التحويل إلى بايت في العمق،فتجميع الذاكرة memory stream لا يعرف إلا ما يخص البايتات فقط.ولكن StreamWriter وStreamReaderيأخذ العبء (الواجب) عنا،يعمل جميع التحويلات المربكة عوضاً عنا.

قراءة ملف بواسطة تجميع(ستريم) Reading a File Via a Stream.

تتضمن معظم المعالجة التابعة لستريم Streamالملفات،لذلك دعنا نستخدم StreamReaderلمعالجة ملف نصي.على الرغم من أننا قررنا سابقاً في الفصل14 أن الملف INIهو شيء من الماضي،من المحتمل أنه من الممتع كتابة روتين يستخرج قيمة من تراث ملف INI.خذ ملف يحتوي هذا النص:

[Section0]
Key1=abc
Key2=def
[Section1]
Key1=ghi
Key2=jkl
[Section2]
Key1=mno
Key2=pqr

والآن يوجد شيء لاتشاهده كل يوم،ولسبب مقنع!مايزال بإمكاننا الحصول على قيمة من أجل Key2في مقطع Section1(القيمة"jkl") إذا ما أردنا ذلك،سيكون علينا الرجوع بالكامل إلى استدعاء GetPrivateProfileStringلواجهة البرمجة التطبيقية APIمن الأيام التي تعود إلى ما قبل الدوت نت السيئة.أو نستطيع تنفيذ StreamReaderفي دالة مخصصة من قبلنا:

```
Public Function GetINIValue(ByVal sectionName As String, ByVal keyName As String, ByVal iniFile As
String) As String
        ' ----- Given a section and key name for an INI file,
        ' return the matching value entry.
        Dim readINI As IO.StreamReader
        Dim oneLine As String
        Dim compare As String
        Dim found As Boolean
        On Error GoTo ErrorHandler
        ' ----- Open the file.
        If (My.Computer.FileSystem.FileExists(iniFile) = False) Then Return ""
        readINI = New IO.StreamReader(iniFile)
        ' ----- Look for the matching section.
        found = False
```

```
        compare = "[" & Trim(UCase(sectionName)) & "]"
        Do While (readINI.EndOfStream = False)
            oneLine = readINI.ReadLine()
            If (Trim(UCase(oneLine)) = compare) Then
                ' ----- Found the matching section.
                found = True
                Exit Do
            End If
        Loop
        ' ----- Exit early if the section name was not found.
        If (found = False) Then
            readINI.Close()
            Return ""
        End If
        ' ----- Look for the matching key.
        compare = Trim(UCase(keyName))
        Do While (readINI.EndOfStream = False)
            ' ----- If we reach another section, then the
            ' key wasn't there.
            oneLine = Trim(readINI.ReadLine())
            If (Len(oneLine) = 0) Then Continue Do
            If (oneLine.Substring(0, 1) = "[") Then Exit Do
            ' ----- Ignore lines without an "=" sign.
            If (InStr(oneLine, "=") = 0) Then Continue Do
            ' ----- See if we found the key. By the way, I'm
            ' using Substring( ) instead of Left( ) so
            ' I don't have to worry about conflicts with
            ' Form.Left in case I drop this routine into
            ' a Form class.
            If (Trim(UCase(oneLine.Substring(0, InStr(oneLine, "=") - 1))) = compare) Then
                ' ----- Found the matching key.
                readINI.Close()
                Return Trim(Mid(oneLine, InStr(oneLine, "=") + 1))
            End If
        Loop
        ' ----- If we got this far, then the key was missing.
        readINI.Close()
        Return ""
ErrorHandler:
        ' ----- Return an empty string on any error.
        On Error Resume Next
        If (readINI IsNot Nothing) Then readINI.Close()
        readINI = Nothing
        Return ""
    End Function
```

هذا الروتين ليس استبدال لـ GetPrivateProfileString بالضبط،فهو لايدعم قيمة راجعة افتراضيةdefault return value، أو عمل تخزين مؤقت مخفي caching للملف من أجل السرعة.تستطيع تحسين الروتين بمعالج خطأ أفضل.ولكنه يعمل على استخراج القيمة التي نبحث عنها،ويعمل ذلك بقراءة الملف INIسطر سطر كل مرة من خلال قارئ التجميع(الستريمStreamReader).

```
MsgBox(GetINIValue("Section1", "Key2", " C:\ini.txt"))
 'jkl' يـعرض        '
```

إدارة الملفات باستخدام فضاء الأسماء "My " File Management with the My Namespace.

يتضمن فضاء الأسماء My عدة ميزات لإدارة الملفات في تفرعه My.Computer.FileSystem، ومن ضمنها ميزات تعمل على إنشاء ستريمات للقراءة والكتابة.

فضاء الأسماء My مقابل أوامر فيجوال بيسك My Namespace Versus Visual Basic Commands.

جميع أعضاء الكائن My.Computer.FileSystem موجودة لاستبدال أو لإكمال ميزات إدارة ملفات موجودة سابقاً في الفيجوال بيسك.يجدول الجدول التالي بعض الميزات الطويلة الموجودة للتفاعل مع الملفات والأدلة في الفيجوال بيسك،وما يكافئها في My.Computer.FileSystem.

ميزة فيجوال بيسك Visual Basic feature	الهدف Purpose	المكافئ My.Computer.FileSystem equivalent
ChDir	تغيير الدليل"العامل"الحالي على المشغل(الدرايف) الافتراضي أو المخصص.	تعطي و تضع الخاصية FileSystem.CurrentDirectory دليل"العمل"الحالي بحيث يفهم من قبل التطبيق.إنك تضع الدليل الفعال من خلال نص المسار المجرد أو المناسب.
ChDrive	تغيير المشغل "العامل" الحالي	لاتعلن فقط الخاصية FileSystem.CurrentDirectory أو تغير الدليل الفعال فهي تعدل أيضاً مشغل(السواقة) الافتراضية.
CurDir	تحديد الدليل"العامل"الحالي والمشغل كنص مسار كامل full path string .	مرة أخرى،FileSystem.CurrentDirectory هي البديل لميزة دليل الفيجوال بيسك.تعمل CurDir على امتلاك مرونة أكبر:فهي تتيح لك تحديد الدليل الحالي على مشغل بدل المشغل الحالي.وهذا لايمكن عمله بواسطة

		.FileSystem.CurrentDirectory
Dir	استخراج ملفات وأدلة في الدليل الرئيسي بحيث يتطابق مع نموذج اسم معين.	تدعم الطرق FileSystem.GetFilesكل من نماذج القيمة الشاملة wildcard عند استخراج دليل مطابقة، وأسماء الملفات.تطلب Dirمنك استدعاءها حالاً من أجل كل مدخلة يجب استعادتها،ولاتعمل بشكل جيد عند معالجة الأدلة المتداخلة.يعود المكافئ FileSystemبمجمعات للبنود المتطابقة،ويمكن بشكل اختياري أن تنحدر كامل شجرة الأدلة الفرعية للمسار القاعدي.
FileCopy	صنع نسخة من ملف.	يوفر FileSystem.CopyFile العديد من الميزات الإضافية القريبة من المستخدم خلفFileCopy .
FileDateTime	استخلاص التاريخ والوقت المنشئ أو المعدل لملف	استخدم الطريقة FileSystem.GetFileInfoلاستخلاص كائن FileInfoالمتخم بالتفاصيل حول ملف ما.من المحتمل أنك ستركز على الخاصية FileInfo.LastWriteTime،ولكن تستطيع الحصول على وقت الإنشاء الأصلي،وقت الوصول الأخير،ميزات غير ممكنة من خلال الدالة FileDateTime.
FileLen	استخلاص الطول بالبايت لملف ما.	الحصول على الكائن FileInfoمن خلال الطريقة FileSystem.GetFileInfo والوصول إلى خاصية Lengthلذلك الكائن للحصول على حجم الملف بالبايتات.
GetAttr	استخلاص المواصفات لملف ما كحقل بت.	الحصول على تفاصيل ملف من خلال الطريقة FileSystem.GetFileInfo، واستخدام الخاصية المعادة Attributesللكائن FileInfoلتفحص مواصفة خيارك.ويعرض هذا الكائن أيضاً القيمة المنطقية IsReadOnly.
Kill	حذف ملف أو دليل فارغ.	تستبدل الطرق FileSystem.DeleteFile و FileSystem.DeleteDirectory الإجراءKill، وتوفر خيارات إضافية غير متاحة من خلال Kill.
MkDir	إنشاء دليل جديد.	إن الطريقة FileSystem.CreateDirectory هي بديل مناسب لـ MkDir.على أية حال، "mkdir" أمر يونيكس قديم،وأنت لست مبرمج على يونيكس أليس كذلك؟
Rename	تغيير اسم ملف أو دليل.	يتم استبدال Renameبالطرق المميزة FileSystem.RenameFile و FileSystem.RenameDirectory .
RmDir	حذف دليل حتى ولو كان يحتوي ملفات.	تحذف الطريقة FileSystem.DeleteDirectory الأدلة التي ما تزال تحوي ملفات أخرى،وهو فعل ما مرفوض من قبل RmDir.يوجد خيار لإرسال الملفات إلى سلة المحذوفاتRecycle Bin .
SetAttr	تعديل مواصفات ملف ما باستخدام حقل بتbit field .	نفس المعالجة المجدولة لـ GetAttrالمذكورة سابقاً في هذا الجدول.الخاصيات Attributesو IsReadOnly لكائن FileInfoهي قراءة/كتابة قيم،على فرض أن لديك حقوق الأمن الضرورية لتغيير مواصفاتattributes .

لماذا تعمل ميكروسوفت على تقديم العديد من ميزات My والتي تضاعف ميزات فيجوال بيسك الموجودة؟ربما هي طريقة لجلب الاستقرار لممارسة البرمجة المعتمدة على ملف من خلال مقاربة أكبر للتوجه الكائني.

قراءة وكتابة ملفات من خلال " My ." Reading and Writing Files Through My

توفر الطرق My.Computer.FileSystem.OpenTextFileReader والموازي OpenTextFileWriter اختصارات لمشيد معتمد على اسم ملف من أجل الكائنات StreamReader وStreamWriter. العبارة:

```
Dim inputStream As IO.StreamReader = My.Computer.FileSystem.OpenTextFileReader(fileNamePath)
```

مطابقة للعبارة التالية:

```
Dim inputStream As New IO.StreamReader(fileNamePath)
```

بالنسبة لي،العبارة الثانية هي الأفضل بسبب طبيعتها المصقولة.إذا أردت تحميل المحتوى الكامل لملف ما ضمن إما مصفوفة بايت أو مصفوفة نصية ليس هناك حاجة لفتح ستريم حالياً بما أن Myيتضمن الطريقة My.Computer.FileSystem.ReadAllTextوالطريقة ReadAllBytes ذات الصلة.هذه العبارة تفرغ المحتوى الكامل لملف ما في سلسلة حرفية.

```
Dim wholeFile As String = My.Computer.FileSystem.ReadAllText(fileNamePath)
```

تعمل كل من الطريقة My.Computer.FileSystem.WriteAllText والطريقة WriteAllBytes نفس الشيء،ولكن باتجاه متعاكس.يوجد إلحاق لمعامل نسبي منطقي(بولين) والذي يتيح لك إما إلحاق أو استبدال محتوى جديد ذو صلة(مناسب) لأي محتوى موجود في ملف.

```
My.Computer.FileSystem.WriteAllText(fileNamePath, dataToWrite, True) ' True=إلحاق
```

ميزة دائماً مفقودة من الفيجوال بيسك وهي القدرة على عمل مسح مناسب على ملف محدد(محدد بالنهايات)(مثل المحدد الجدولي tab-delimited أو محدد بالفاصلة-comma-delimited)أو ملف حقل ثابت الاتساعfixedwidth-field، واستخراج الحقول على كل سطر دون الحاجة إلى كود تفسير(ترجمة)إضافيextra parsing code .
تتضمن الفيجوال بيسك الآن الكائن Microsoft.VisualBasic.FileIO.TextFieldParser الذي يبسط هذه المعالجة(العملية).وهذا الكائن يتيح لك الإشارة إما إلى محدد الحقل(مثل رمز الجدولةtab)أو مصفوفة سعات عمودcolumn sizes . حالما تصحبه بمسار ملف،فإنه يقرأ كل سطر بيانات،يقسم الحقول المميزة لك ضمن مصفوفة نصية.
تفتح الطريقة My.Computer.FileSystem.OpenTextFieldParser الملف وتعرف طريقة الترجمة في نفس الوقت.

```
Dim sourceFile As FileIO.TextFieldParser
```

```
        ' ----- Open the file with tab-delimited fields.
        sourceFile = My.Computer.FileSystem.OpenTextFieldParser(sourceFilePath, vbTab)
        ' ----- Process each line.
        Do While Not sourceFile.EndOfData
            dataFields = sourceFile.ReadFields()
            ' ----- dataFields is a simple string array,
            ' so you can examine each field directly.
            If (dataFields(0) = "NEW") Then
                ' ----- and so on...
Loop
        sourceFile.Close()
```

يمكن للكائن TextFieldParserأيضاً أن يكتشف أسطر التعليق ويتجاهلها بشكل صامت.

مشروع.Project

لدي بعض الأخبار السيئة والأخبار الجيدة،الأخبار السيئة هي أن مشروع المكتبة لايعمل قراءة أو كتابة مباشرة للملفات القياسية،وليس لديه حاجة لستريمات ملف.وهذا يعني أننا لن نعمل على إضافة أي كود إلى المشروع في هذا الفصل على الإطلاق.الأخبار الجيدة ستبقى لدينا أشياء مهمة للتحدث حولها.بالإضافة أننا تجاوزنا أكثر من نصف الكتاب،فيمكنك أخذ راحة.

تركيب(تشكيل)مخرجات سجل.Configuring Log Output

متى حدث خطأ في تطبيق المكتبة،يعمل الروتين أولاً على إظهار رسالة الخطأ للمستخدم،ومن ثم يسجلها إلى أي "منصتات التسجيل log listeners".

```
Public Sub GeneralError(ByVal routineName As String, ByVal theError As System.Exception)
'        إعلام المستخدم بـالخطأ
        On Error Resume Next
        MsgBox("حدث الخطأ التـالي عند الموقع التـالي '" & routineName & "':" & vbCrLf & vbCrLf & 
theError.Message, MsgBoxStyle.OkOnly Or _
        MsgBoxStyle.Exclamation, ProgramTitle)
        My.Application.Log.WriteException(theError)
    End Sub
```

لذلك،من ينصت؟إذا شغلت التطبيق ضمن الفيجوال أستوديو،فإن الفيجوال بيسك تركب دائماً منصت تسجيل يعمل على عرض النص في لوحة النافذة المباشرة Immediate Window.ولكن هذا لايقوم بعمل الكثير من العمل الجيد في التطبيق المترجم أو الموزع.تستطيع تصميم منصتاتك الخاصة،ولكن تتضمن أيضاً الدوت نت العديد من المنصتات المعرفة مسبقاً،يمكن تمكينها جميعاً وتركيبها من خلال ملف التطبيق*app.config*. إذا راجعت مشروع الفصل14 ستجد محتوى الملف *app.config*والذي يعمل على إعداد واحد منها كمنصت.

العموميات(الشموليات).Generics

تستطيع الوصول إلى العمومياتgenerics ،من خلال تقنية العموميات للدوت نت.العموميات:هي القدرة على استخدام "حافظات أمكنةplaceholders "لأنواع البيانات-أول ما ظهرت في الفيجوال بيسك2005 وإطار عمل الدوت نت2.0 ذو الصلة.يقدمك هذا الفصل إلى "النوعياتspecifics " على العموميات.

ما هي العموميات؟.What Are Generics?

في الدوت نت"العموميات"هي تقنية تتيح لك تعريف حافظات placeholdersأنواع بيانات ضمن أنواع أو طرق.لنقول أنك احتجت إلى تعريف فئة تتعقب بيانات زبون،ولكن لاتريد أن تنفذ تنسيق معين على قيمة "المعرفID" للزبون.يحتاج قسم من كودك التفاعل مع كائنات الزبون باستخدام قيمة معرف ID انتغر،بينما سيستخدم جزء آخر من الكود مفتاح ترقيم حرفي(أبجدي) من أجل الزبون.ربما تتسآل،لما لا تعمل على تضمين فقط كلاً من نوعي المعرف كحقول مميزة في سجل الزبون؟ إن هذا لن يعمل لأنني أحاول إيجاد حل لمثال بسيط معقول نوعاً ما والإجابة عن ذلك السؤال ستحيرني.لذلك إليك النسخة العددية من الفئة:

```
Class CustomerWithNumberID
        Public ID As Integer
        Public FullName As String
    End Class
```

وهنا التنوع الذي يستخدم معرف ID نصي.

```
Class CustomerWithStringID
        Public ID As String
        Public FullName As String
    End Class
```

بالطبع تستطيع تعريف IDككائنSystem.Object، وتُلصِق stickأي شيء تريده في ذلك الحقل.ولكن يتم اعتبار System.Object"محدد النوع بشكل ضعيف weakly typed "، ولا يوجد شيء يوقفك من المزج بين قيم المعرف الإنتغر Integerوالنصية Stringمن أجل حالات مختلفة في مصفوفة لكائنات زبون.
ما تحتاجه هو نظام يتيح لك تعريف الفئة بشكل عمومي،وتنأى بنفسك عن تعين نوع البيانات للمعرف IDحتى تنشئ بشكل عملي حالة من الفئة،أو تجمع كامل من حالات الفئة ذات الصلة.مع نظام مثل هذا،بإمكانك تعريف(تحديد)نسخة الأهداف العامة لفئة الزبون.

```
Class CustomerWithSomeID
Public ID As <DatatypePlaceholder>
    Public FullName As String
End Class
```

فيما بعد عندما يحين الوقت لإنشاء حالة ،تستطيع إخبار اللغة بنوع البيانات الذي سيستخدم لـ"حاجز المكانplaceholder ".

```
Dim oneCustomer As CustomerWithSomeID(replacing <DatatypePlaceholder> with Integer)
```

هذا ما تتيح لك الشموليات عمله.قواعد الفيجوال بيسك التي تعرف(تحدد) فئة الزبون الغير معينة(الغير مخصصة):

```
Class CustomerWithSomeID(Of T)
        Public ID As T
        Public FullName As String
    End Class
```

حاجز المكان العام،T ،يظهر في شرط Ofالخاص،تماماً بعد اسم الفئة،(ليس عليك تسمية حافظ المكان *T*،ولكن أصبح تقليد عند تقديم كود بسيط يستخدم العموميات generics.)كنوع بيانات،يمكن أن يتم استخدامT في أي مكان ضمن تعريف الفئة حيث أنك لا تريد تحديد نوع البيانات مقدماً.الفئة وعضوهاID،هما الآن جاهزان للاستنساخ مع نوع بيانات بديل حقيقي من أجل *T*. لإنشاء حالة جديدة ،جرب الكود:

```
Dim numberCustomer As CustomerWithSomeID(Of Integer)
```

عندما يتم إرفاق (Of Integer)إلى نهاية تعريف فئة،تتصرف الفيجوال بيسك وكأنك صرحت عملياً عن متغير لفئة لديها عضو انتغر مسمى ID.في الحقيقة،إنك تفعل.عندما تعمل على إنشاء حالة من فئة عمومية،يعرف المترجم فئة منفصلة تبدو فئة غير عمومية مع تبديل جميع حاجزات الأمكنة.

```
Dim customer1 As New CustomerWithSomeID(Of Integer)
Dim customer2 As New CustomerWithSomeID(Of Integer)
Dim customer3 As New CustomerWithSomeID(Of String)
```

تعرف هذه الأسطر حالتين من CustomerWithSomeID(Of Integer)، وحالة من CustomerWithSomeID(Of String) . customer1 وcustomer2 هي حالات حقيقية من نفس نوع البيانات،ولكن customer3 هو حالة لنوع بيانات مختلف تماماً. الإسنادات بين customer1 وcustomer2ستعمل،ولكن لا تستطيع مزج أياً منهما مع customer3دون عمل تحويل صريح.

```
'هذا سيعمل بشكل جيد
        customer1 = customer2
 '           هذا لن يعمل
        customer3 = customer1
```

كحقيقة،أنواع بيانات وقت الترجمة المولدة بشكل آلي بواسطة المترجم، تعرض جميع ذاتيات الفئات الغير عمومية.حتى المساعد الفوري IntelliSenseبالمعنى الضيق للكلمة يكتشف نوع البيانات البديل.يتضمن الشكل التالي تلميح أداة tool tip ،تماماً على يمين قائمة اختيار عضو الحالة،والتي تعرف بالضبط العضو customer1.ID كإنتغر Integer .

```
Dim customer1 As New CustomerWithSomeID(Of Integer)
Dim customer2 As New CustomerWithSomeID(Of Integer)
Dim customer3 As New CustomerWithSomeID(Of String)

customer1.|
```

Equals
FullName
GetHashCode
GetType
ID — Public ID As Integer
ReferenceEquals
ToString
Common | All

ضمن تعريف الفئة،يمكن لحاجز المكان T أن يظهر في أي مكان،حتى ضمن قوائم معاملات نسبية argument وتصريحات متغيرات محلية.

```
Class SomeClass(Of T)
    Public Function TransformData(ByVal sourceData As T) As T
' أضف كود تحويل العموميات هنا.
        Dim workData As T
...
    End Function
End Class
```

تعمل العموميات مع التراكيب والواجهات أيضاً.

```
Structure SomeStructure(Of T)
        Public GenericMember As T
    End Structure
    Interface ISomeInterface(Of T)
        Sub DoWorkWithData(ByVal theData As T)
    End Interface
```

تنوعات(اختلافات) للتصريح العمومي.Variations of Generic Declaration

إذا كان هناك أقل حاجة لحاجز مكان نوع بيانات حكومة اتحادية federal government، فإن تنفيذ العموميات الموصوف تماماً سيستقبلها بشكل خاص.ونوعها متقن لتأجيل تعريف نوع البيانات حتى أخر دقيقة.ولكن عموميات الدوت نت لا تتوقف هناك.

حاجزات الأمكنة المتعددة.Multiple Placeholders

حاجزات الأمكنة العمومية-وتعرف أيضاً كوسيطات نوع *type parameters* –فكل فئة عمومية يمكن أن تتضمن حاجزات أمكنة متعددة بإضافتهم إلى شرط Of الأولي.

```
Class MultiTypes(Of T1, T2)
        Public Member1 As T1
        Public Member2 As T2
    End Class
```

كما ذكرت من قبل،ليس عليك استخدام الأسماء المملة T1وT2 .مهما تكن الأسماء التي تختارها،ضمنها كقائمة مفصولة بفاصلة تماماً بعد الكلمة المحجوزة Of.عندما تكون جاهز لإنشاء حالة ،كرر القائمة المحددة بفاصلة في نفس الترتيب،ولكن استخدم نوع حقيقي.في هذه العبارة،يحل انتغر Integer محل T1 ونص String محل T2.

```
Dim useInstance As MultiTypes(Of Integer, String)
```

قيود نوع البيانات والوسيطات(الواجهات).Data Type and Interface Constraints

نوع الوسيطات التي تضمنها في عمومي ما،مثل T،يقبل أي نوع بيانات صحيح(محقق)ومن ضمنها الانتغر،السترينغ(السلة الحرفية)،نماذج ويندوز System.Windows.Forms.Form، أو أنواع خاصة بك. يمكن أن يتم استبدال T بأي شيء تشتقه من System.Object،والذي هو كل شيء،تستطيع حتى تصور العبارة.

```
Class SomeClass(Of T)
```

يمكن تبديلها بما يلي:

```
Class SomeClass(Of T As System.Object)
```

إضافة الشرط Asلجعلها تبدو مثل تصريحات الفيجوال بيسك الأخرى.حسناً،تستطيع التوقف عن التخيل وتبدأ العمل:تدعم حاجزات الأمكنة الشرط As.لو لم تضمن الشرط As، تعتبر الفيجوال بيسك أنك تعني System.Object، ولكن بإمكانك إتباع As بأي نوع تريد:

```
Class FormOnlyClass(Of T As System.Windows.Forms.Form)
...
End Class
```

بإضافة فئة معينة مع الشرطAs، فإنك تطبق قيد *constraint* على النوع العمومي،حد limitation يجب أن يتم مطابقته لاستخدام النوع.في هذه الحالة، يقول القيد:"يمكن أن توفر أي قيمة لـ *T* طالما أنه أو مشتق من System.Windows.Forms.Form."وهذا يعني أنك تستطيع إنشاء حالة من FormOnlyClass باستخدام واحد من نماذج تطبيقك،ولكن ليس باستخدام فئات غير الفورم.

```
' هذا سيعمل.
        Dim usingForm As FormOnlyClass(Of Form1)
' هذا لن يعمل.
```

```
        Dim usingForm As FormOnlyClass(Of Integer)
```

عندما تضيف قيد لوسيط نوع،فإنه يؤثر في الميزات التي بإمكانك استخدامها مع وسيط النوع.خذ هذه الفئة العمومية المقرر لها العمل مع النماذج،ولكن غير مصرح عنها بتلك الطريقة:

```
Class WorkWithForms(Of T)
    Public Sub ChangeCaption(ByVal whichForm As T, ByVal newCaption As String)
'            السطر التالي لن يتم ترجمته
        whichForm.Text = newCaption
    End Sub
End Class
```

في هذه الفئة ،سيفشل الإسناد إلى whichForm.Text لأن الفئة WorkWithForms لاتعرف أنك تخطط لاستخدامها مع النماذج.فما تعلمه أنك تخطط لاستخدام T،و T،بشكل افتراضيي من نوع System.Object.ولا توجد خاصية نص Textفي الفئةSystem.Object .

إذا غيرت تعريف الفئة WorkWithFormsلأن تقبل الكائنات Form،فإن وجهة نظر ترجمة هذا الكود تتغير بشكل مثير.

```
Class WorkWithForms(Of T As Windows.Forms.Form)
        Public Sub ChangeCaption(ByVal whichForm As T, ByVal newCaption As String)
'سيترجم الآن.
            whichForm.Text = newCaption
        End Sub
    End Class
```

بما أن T عليه أن يكون من نوع Formأو شيء مشتق من نموذج،فإن الفيجوال بيسك تعرف جميع أعضاء فئة الفورم،ومن ضمنها النصText، فهو الآن متاح لجميع أشياء T.وبالتالي يعمل الإسناد whichForm.Text.

بالإضافة إلى الفئات،تستطيع أيضاً استخدام الواجهات interfacesلتقيد أنواعك العمومية.

```
Class ThrowAwayClass(Of T As IDisposable)
```

فالحالات من ThrowAwayClassيمكن أن يتم إنشاءها عند الحاجة،ولكن إذا تم تزويد النوع فقط مع التصريح الذي ينفذ الواجهة IDisposable.

```
'هذا سيعمل. بما أن الأقلام تستخدم:
        'IDisposable
        Dim disposablePen As ThrowAwayClass(Of System.Drawing.Pen)
 '         لن يعمل التالي,بما أن نوع بيانات الانتغر لاتنفذ:
        'IDisposable
        Dim disposableNumber As ThrowAwayClass(Of Integer)
```

ولكن تستطيع أيضاً إتباع الشرط Asعلى حافظة العمومي بالكلمة المحجوزة New.

```
Class SomeClass(Of T As New)
End Class
```

يقول الشرط As New لنوع العمومي،"اقبل أي نوع لـ T،ولكن فقط إذا تضمن ذلك النوع مشيد لا يتطلب معاملات نسبية."وبالتالي على T أن يتضمن مشيد افتراضيي.حالما يتم تعريفه،ستكون قادر على إنشاء حالات جديدة من T ومهما يكن النوع فإنه يحدث لأن يكون في نوعك العمومي.

```
Class SomeClass(Of T As New)
    Public Sub SomeSub()
        Dim someVariable As New T
    End Sub
End Class
```

إذا تضمنت فئتك العمومية وسيطات لأنواع متعددة،فكل وسيط يمكن أن يتضمن الشرط Asالخاص به مع النوع المميز أو تقيد الواجهة interface constraint .

القيود المتوافقة(المتزامنة). Simultaneous Constraints

أحياناً تكون بحاجة إلى أداة متعددة الوظائف،كما رأيت في كل حافظة عمومية.إذا كنت بحاجة إلى حافظة واحدة لتضمين قيد لفئة معينة class، واجهة interface،و"جديد New " كلها مع بعضها البعض(أو فجأةً)تستطيع فعل هذا.اعمل على تضمين بعد الكلمة المحجوزة Asعدة قيود في أقواس مجعدة:

```
Class SomeClass(Of T As {Windows.Forms.Form, IDisposable, New})
End Class
```

والآن أي نوع توفره في الشرط *of* عند إنشاء حالة من هذه الفئة يجب أن يلاقي جميع القيود*constraints*، وليس واحد منها.وإليك شيء جديد:تستطيع تضمين أكثر من قيد واجهة واحد في نفس الوقت.

```
Class SomeClass(Of T As {Runtime.Serialization.ISerializable, IDisposable})
End Class
```

والآن مايزال بإمكانك تضمين تقيد فئة classوتقيد"جديد New "،حتى مع هذه الواجهات المتعددة.(تستطيع تضمين أكثر من تقيد فئة واحدة من أجل حافظة مفردة).لو تضمن نوعك العمومي وسيطات أنواع متعددةmultiple type parameters، كل منها يمكن أن يكون لديه مجموعة القيود المتعددة الخاصة به.

تداخل أنواع العمومي.Nesting Generic Types

يمكن لأنواع العمومي أن تتضمن أنواع متداخلة خاصة بها.

```
Class Level1(Of T1)
    Public Level1Member As T1
    Class Level2(Of T2)
        Public Level2Member1 As T1
        Public Level2Member2 As T2
    End Class
End Class
```

فيمكنك مداخلة العموميات بالعمق الذي تريد .

Non-Generic Types with Generic Members.الأنواع الغير عمومية مع أعضاء عمومية

إذا بدت الأنواع العمومية مخيفة أو ساحقة قليلاً،لا تغتاظ.فليس عليك إنشاء نوع عمومي بالكامل لاستخدام ميزات العمومي الجديدة.تستطيع إضافة دعم عمومي لطريقة وحيدة فقط ضمن فئة عادية مختلفة.

```
Class SomeClass
' الفئة نفسها ليس لديها عمومي أو شرط أوف,لذلك هي ليست شمولية ولكن التالي:
    Public Shared Sub ReverseValues(Of T)(ByRef first As T, ByRef second As T)
' هذه الطريقة هي عمومية أو شمولية مع شرطها الخاص أوف:
' اعكس محتويات كل من المتغيرين
        Dim holdFirst As T
        holdFirst = first
        first = second
        second = holdFirst
    End Sub
End Class
```

إن طرق العمومي Genericمفيدة عندما تحتاج متغير محلي من نوع الحافظة placeholder ضمن طريقة(كما عملنا مع holdFirst هنا)،ولكن لا تعرف النوع مقدماً.استخدام هذه الطريقة ReverseValues المتشاركة sharedيعمل مثل الطرق الأخرى،مع شرط الزائد Of.

```
Dim x As Integer = 5
Dim y As Integer = 10
SomeClass.ReverseValues(Of Integer)(x, y)
' يعرض 10
MsgBox(x)
```

إذا كنت ستستخدم الحافظة لواحد أو أكثر من معاملات طريقة،ستعمل الفيجوال بيسك على استنتاج النوع بالاعتماد على القيمة الممررة.إذا كان بإمكان الفيجوال بيسك تخمين النوع في هذه الطريقة،فلا تحتاج حتى إلى الشرط Ofعند استدعاء طريقة العمومية.

```
SomeClass.ReverseValues(x, y)
```

كما مع أنواع العمومي،تسمح لك طرق العمومي إضافة قيود إلى الحافظات.

Overloading Generic Types and Members.إعادة تعريف أنواع وأعضاء العمومي

ذكرت سابقاً كيف يعمل المترجم على إنشاء فئات مستقلة بشكل أساسي لكل تنوع حالة لفئة عمومية قمت بإنشائها.وهذا يعني أن كل من هاتين الحالتين تستخدمان بالفعل وبشكل عام فئات مستقلة ومختلفة بالكامل:

```
Dim numberVersion As SomeClass(Of Integer)
Dim textVersion As SomeClass(Of String)
```

حيث أن كل من SomeClass(Of Integer) وSomeClass(Of String)فئتين مختلفتين بالكامل،حتى ولو كان لديهما نفس الاسم القاعدي.بطريقة ما،تعمل الفيجوال بيسك على إعادة تعريف اسم الفئة من أجلك،بحيث تتيح لك استخدامها بطريقتين مختلفتين أو أكثر.

تتيح لك العموميات أيضاً تضمين اسم معطى(أو ثابت) في لعبة إعادة تعريف فئة.عادةً،تستطيع إنشاء فقط فئة وحيدة مع اسم ثابت ضمن فضاء أسماء خاص،ولكن مع العموميات،تستطيع إعادة استخدام اسم فئة،طالما أن استخدام الحافظات بين الفئات مختلف تماماً،إما في عددها أو في قيودها المطبقة.

```
Class SomeClass(Of T1)
' هذه فئة عمومية مع حافظة وحيدة
    End Class
    Class SomeClass(Of T1, T2)
' هذه فئة عمومية مختلفة بالكامل مع حافظتين.
    End Class
```

ستعمل الفيجوال بيسك على تقرير أي نسخة من الفئتين ستستخدم بالاعتماد على الشرط Ofالذي عملت على تضمينه مع تصريح الحالة.

```
Dim simpleVersion As SomeClass(Of Integer)
Dim complexVersion As SomeClass(Of Integer, String)
```

Generics and Collections.العموميات والتجمعات

في الحقيقة تشع العموميات في منطقة التجمعات.الإصدار الأول من الدوت نت لديه من بين الآلاف من الفئات الممكنة المفيدة،مجموعة من فئات"التجمع collection"،جميعها في فضاء الأسماء System.Collections. يتيح لك كل تجمع حشو أي حالات لكائنات أخرى كما تريد داخل ذلك التجمع،واستخراجها فيما بعد.تختلف التجمعات في كيفية الحشو والاستخراج،ولكنها جميعاً تتيح لك إلصاق أي نوع من الكائنات في التجمع.

واحدة من فئات التجمع هي الفئة System.Collections.Stack.تتيح لك المكدسات Stacksتخزين كائنات بشكل منضد:الكائن الأول الذي تعمل على إضافته إلى المكدس يذهب إلى الأسفل وكل واحد تعمل على إضافته يذهب إلى أعلى الكائن السابق له.وعندما تعمل على إزالة بند ما،فإنه يزول من الأعلى.(هذا نظام"ما يدخل أخيراً،يخرج أولاً last in, first out "وفي بعض الأحيان يدعى " LIFO ") تدير الطريقة Pushوالطريقة Popإضافة وإزالة الكائنات.

```
Dim numberStack As New Collections.Stack
        numberStack.Push(10)
        numberStack.Push(20)
        numberStack.Push(30)
        MsgBox(numberStack.Pop()) ' Displays 30
        MsgBox(numberStack.Pop()) ' Displays 20
        MsgBox(numberStack.Pop()) ' Displays 10
```

توجد أيضاً الطريقة Peekوالتي تعتبر البند الأعلى،ولكنها لاتزيله من المكدس.الأمر مع المكدسات(والتجمعات المشابهة الأخرى)هو أنه ليس عليك وضع فقط نوع واحد من الكائنات ضمن المكدس.تستطيع مزج جميع أنواع الكائنات التي تريدها.

```
Dim numberStack As New Collections.Stack
        numberStack.Push(10) ' Integer
        numberStack.Push("I'm sneaking in.") ' String
        numberStack.Push(Me.Button1) ' Control
```

لايبالي المكدس،بما أنه يعامل كل شيء كـ System.Object.ولكن ما الذي يحدث إذا ما احتجت لضمان أن يتم وضع انتغر *integers* فقط ضمن مكدس؟ ما الذي يحدث إذا ما أردت تحديد(أو تخصيص)مكدس ما إلى نوع بيانات معين،ولكن لا تريد كتابة فئات مكدس منفصلة(أو مستقلة)لكل نوع ممكن؟

من المؤكد أن هذا يبدو لي كعمل العموميات بالنسبة لي.من الواضح أنها طريقة ميكروسوفت،أيضاً،لذلك،عملت على إضافة تفرع لتجمعات عمومية جديدة إلى إطار العمل.وهي تظهر في فضاء الأسماء System.Collections.Generic. توجد عدة فئات مختلفة في فضاء الأسماء هذا.متضمناً فئات للقوائم المرتبطة linked lists،الطوابير queues، والقواميس dictionaries. توجد فئة تدعى Stack(Of T). وهي ما نريد تماماً.

```
Dim numberStack As New Collections.Generic.Stack(Of Integer)
        numberStack.Push(10)
        numberStack.Push(20)
        numberStack.Push(30)
```

والآن إذا حاولنا إضافة أي شيء غير الانتغر إلى numberStack،سيحدث خطأ فالكود التالي لن يعمل:

```
numberStack.Push("I'll try again.")
```

أنواع "قابلة بدون قيمة Nullable"العمومية(الشمولية).Generic Nullable Types

بالعودة إلى الفصل 6،فقد قدمت الأنواع" القابلة بدون قيمة nullable "،وهي طريقة السماح لأن يتم استخدام"لاشيء Nothing " مع الأنواع ذات القيمة value types .

```
Dim numberOrNothing As Integer?
```

على الرغم من أنك لاتستطيع الحكم من سطر الكود المصدري هذا،فالأنواع بدون قيمة nullable يتم تنفيذها بالفعل باستخدام العموميات. نسخة التصريح الكاملة لـ numberOrNothing's هي:

```
Dim numberOrNothing As Nullable(Of Integer)
```

تعمل الفيجوال بيسك على توفير وبكل بساطة اختصار لهذه القاعدة من خلال اللاحقة؟ .تستطيع استخدام أياً من التركيبين للتصريح عن الحالات القابلة لأن تكون بدون قيمة nullable .

مشروع. project

عندما يستخرج زبون كتاب أو بند مكتبة آخر،يتم حساب تاريخ استحقاق الدفع بشكل آلي بالاعتماد على عدد الأيام المخزنة في حقل قاعدة البيانات CodeMediaType.CheckoutDays. ولكن ما الذي يحدث لو كان التاريخ المحسوب يوم عطلة،والمكتبة مغلقة؟يمكن أن لا يكون الزبون قادر على إعادة الكتاب حتى اليوم التالي،مما يسبب غرامة.وهذه الغرامة،حتى لو كانت صغيرة، يمكن أن تبدأ سلسلة ردود أفعال في حياة الزبون. لحسن الحظ يمكن تجنب هذا بإضافة قائمة العطل إلى المشروع.فإذا وقع يوم إعادة البند على عطلة موثقة،فإن البرنامج يضبط التاريخ للأمام حتى يجد تاريخ غير العطلة.

إدارة العطل.Managing Holidays

بقدر صغر التطبيق القائمة بذاته والذي يدير بياناته بالكامل،فليس هناك حاجة ملحة للعموميات في تطبيق المكتبة.مهما يكن،توفر الشموليات فوائد أكثر من تقيد أنواع البيانات المخزنة في فئة أو تجمع.فهي تحسن أيضاً دعم المساعد الفوري وتحويل البيانات،الذي يمكن للفيجوال بيسك الإخبار به مباشرةً،على سبيل المثال،ما هو نوع البيانات الذي سيظهر في تجمع.سنعمل على تخزين العطل المدارة بواسطة مشروع المكتبة في جدول Holiday لقاعدة البيانات.محتويات هذا الجدول نادراً ما تتغير،سنخزن البيانات داخل التطبيق.ولتبسيط إدارة تلك البيانات المخزنة،سنعمل على تخزين العطل في تجمع شمولي(عمومي).

أولاً ،دعنا نعمل على إنشاء فئة تحفظ مدخلة عطلة مفردة.أضف فئة جديدة إلى المشروع من خلال قائمة مشروع Project >>إضافة فئة Add Class، وامنحها الاسم HolidaySet.*vb*. يظهر التركيب المعروف للفئة الفارغة.

```
Public Class HolidaySet
End Class
```

يتضمن جدول قاعدة البيانات حقلين رئيسيين يتم استخدامهما لحساب العطل: EntryTypeو EntryDetail.دعنا نخزنهما كعضوين للفئة ،ونضيف إشارة تكفل أن تكون المدخلة صحيحة.

```
Private HolidayType As String
Private HolidayDetail As String
Private IsValid As Boolean
```

سنعمل على ملئ هذه الحقول الخاصة من خلال مشيد الفئة.

```
Public Sub New(ByVal entryType As String, ByVal entryDetail As String)
        ' إنشاء مدخلة نسخة مدخلة عطلة جديدة
        HolidayType = Left(Trim(UCase(entryType)), 1)
        HolidayDetail = entryDetail
        ' لنرى فيما إذا كانت التفاصيل صحيحة
        IsValid = True
        Select Case HolidayType
            Case "A"
                ' التفاصيل يجب أن تكون بالتنسيق
                'mm/dd
                IsValid = IsDate(entryDetail & "/2004")
            Case "E"
                ' التفاصيل رقم من 1 إلى 7
                If (Val(entryDetail) < 1) Or (Val(entryDetail) > 7) Then IsValid = False
            Case "O"
                ' يجب أن تكون التفاصيل تاريخ صحيح
                IsValid = IsDate(entryDetail)
```

```
            Case Else
' الغير صحيح يجب أن لا يحدث
                IsValid = False
        End Select
    End Sub
```

من الواضح،أن مدخلات العطل لديها نظام تشفير خاص بها،ولن يكون من العدل إجبار الكود في مكان آخر من التطبيق لأن يتعامل مع جميع تعقيدات مقارنات تاريخ العطل،لذلك،دعنا نضيف طريقة عامة إلى الفئة تشير فيما إذا كان تاريخ معطى يتوافق مع عطلة مخزنة في حالة.

```
Public Function IsHoliday(ByVal whatDate As Date) As Boolean
' تقديم تاريخ,لنرى هل يطابق نوع المدخلة في هذه النسخة
        Dim buildDate As String
' إذا كان هذا السجل غير صحيح,عندئذٍ فإنها لن تطابق أو توافق عطلة
        If (IsValid = False) Then Return False
        Select Case HolidayType
            Case "A"
                ' سنوي
                buildDate = HolidayDetail & "/" & Year(whatDate)
                If (IsDate(buildDate)) Then
                    Return CBool(CDate(buildDate) = whatDate)
                Else
' يجب أن يكون شباط 29 إذا كانت السنة كبيسة
                    ' non-leap-year.
                    Return False
                End If
            Case "E"
' يوم من أسبوع
                Return CBool(Val(HolidayDetail) = Weekday(whatDate, FirstDayOfWeek.Sunday))
            Case "O"
' لنرى فيما إذا كانت هذه مطابقة لمرة واحدة بالضبط
                Return CBool(CDate(HolidayDetail) = whatDate)
        End Select
    End Function
```

لقد انتهينا من هذه الفئة.والآن ما نحتاج إليه فقط مكان لحفظ سجلات العطل المخزنة.يتضمن فضاء الأسماء System.Collections.Generic عدة فئات تجمع مختلفة نستطيع استخدامها.بما أن الشيء الوحيد الذي نحتاج إلى عمله مع العطل حالما تكون في التجمع هو عمل مسح عليها،البحث عن التطابقات،إن الفئة(List(Of T القياسية تبدو الأفضل،وميزاتها الرئيسية طبقاً لتوثيق الدوت نت،تتيح لك الوصول إلى الأعضاء من خلال الفهرسindex .

افتح ملف *General.vb*واعمل على إيجاد أين تظهر المتغيرات العامة،في مكان ما قريبة من أعلى الفئة.ومن ثم أضف تعريف من أجل تجمع عام سيعمل على تخزين جميع العطل.

```
Public AllHolidays As Collections.Generic.List(Of Library.HolidaySet)
```

هذا تجمع شمولي.

اعمل على إيجاد الطريقةInitializeSystem، مانزال في الملف*General.vb*، أضف الكود التالي والذي سيمهد لتخزين العطل العامة.

```
AllHolidays = New Collections.Generic.List(Of HolidaySet)
```

هذه هي البنية التحتية،لنعمل على إضافة روتينات تتمكن من الوصول إلى قائمة الشمولي هذه.نحتاج إلى روتين سيخبرنا،صواب أو خطأ،فيما إذا تاريخ معطى(تاريخ استحقاق الدفع المخطط له بالنسبة لبند المكتبة)يتطابق مع من العطل أو لا.أضف الدالة IsHolidayDateإلى الملف *General.vb*.

```
Public Function IsHolidayDate(ByVal whatDate As Date) As Boolean
        ' لنرى هل التاريخ المعطى عطلة
        Dim oneHoliday As Library.HolidaySet
        ' إجراء البحث على العطل،للبحث عن تطابق العطل
        For Each oneHoliday In AllHolidays
            If (oneHoliday.IsHoliday(whatDate)) Then Return True
        Next oneHoliday
        ' ليس عطلة
        Return False
    End Function
```

يبين هذا الروتين IsHolidayDate كيف تأتي العموميات مناسبة للاستخدام.فكل السحر يحدث في العبارة For Each.في تجمع عادي،لن نكون واثقين من نوع البند الذي تم تخزينه في التجمع.فيمكن أن تكون HolidaySetأو سترينغ Stringأو انتغرInteger. نحن سنعلم بما أننا المطورين،ولكن الفيجوال بيسك تعبث من غير بصيرة في هذه المنطقة ، و تفترض أنك مزجت أنواع بيانات في تجمع واحد.ولكن لأننا ربطنا التجمع AllHolidaysإلى الفئة HolidaySetباستخدام الشرط Of HolidaySet،فإن الفيجوال بيسك تفهم الآن أننا سنعمل على تخزين فقط بنود HolidaySetفي التجمعAllHolidays. وذلك يعني أنه ليس علينا بشكل صريح تحويل البنود المستخرجة من التجمع إلى نوع بيانات HolidaySet.إذا لم نستخدم فئة العمومي،سيبدو الكود نوعاً ما مشابه لهذا:

```
Dim scanHoliday As System.Object
Dim oneHoliday As LIBRARY.HolidaySet
        For Each scanHoliday In AllHolidays
            oneHoliday = CType(scanHoliday, LIBRARY.HolidaySet)
            If (oneHoliday.IsHoliday(whatDate)) Then Return True
        Next
```

بما أن التجمع العادي يعمل على تلخيص كل شيء إلى System.Object،فسيكون علينا بشكل صريح تحويل كل كائن تجمع إلى HolidaySet باستخدام الدالة CType أو دوال التحويل المشابهة.ولكن مع تجمع العمومي،تأخذ الفيجوال بيسك العبء عنا. سنبقى بحاجة إلى تخزين العطل من قاعدة البيانات،لذلك أضف الطريقة RefreshHolidays إلى *General.vb* والتي تقوم بهذا العمل.

```
Public Sub RefreshHolidays()
        ' التحميل بقائمة العطل.
        Dim sqlText As String
        Dim dbInfo As SqlClient.SqlDataReader
        Dim newHoliday As Library.HolidaySet
        On Error GoTo ErrorHandler
        ' إزالة القائمة الحالية من العطل
        AllHolidays.Clear()
        ' الحصول على العطل من قاعدة البيانات
        sqlText = "SELECT * FROM Holiday"
        dbInfo = CreateReader(sqlText)
        Do While dbInfo.Read
            newHoliday = New Library.HolidaySet(CStr(dbInfo!EntryType), CStr(dbInfo!EntryDetail))
            AllHolidays.Add(newHoliday)
        Loop
        dbInfo.Close()
        Return
ErrorHandler:
        GeneralError("RefreshHolidays", Err.GetException())
        On Error Resume Next
        If Not (dbInfo Is Nothing) Then dbInfo.Close() : dbInfo = Nothing
        Return
    End Sub
```

لقد رأيت الكثير من الكود مشابه لهذا سابقاً،الكود الذي يعمل على تحميل سجلات من جدول قاعدة البيانات ضمن البرنامج.

يوجد مكانين حيث نحتاجهما لاستدعاء الطريقة RefreshHolidays: عندما يبدأ البرنامج للمرة الأولى،وفيما بعد حيثما تحدث تغيرات إلى قائمة العطل.لن نقلق فيما يخص التغيرات على القائمة من قبل المستخدمين الآخرين،سنركز فقط على متى يعمل التطبيق المحلي على تحديث القائمة.أولاً،افتح الملف *ApplicationEvents.vb*،أضف الكود التالي إلى معالج حدث MyApplication_Startup،بعد الاستدعاء الموجود لـ LoadDatabaseSettings( ) .

```
RefreshHolidays()
```

كما ترى تماماً في هذا الروتين،لقد أضفنا سابقاً المحرر الذي يدير قائمة العطل.الشيء الوحيد الباقي عمله هو الوصول الحقيقي إلى قائمة العطل عند استخراج البنود.وسنعمل هذا في الفصول اللاحقة إن شاء الله.

الاستعلام المتكامل للغة LINQ

لقد تعززت فيجوال بيسك الآن بلينكو،ميزة جديدة في الفيجوال بيسك2008 تتيح لك الاستعلام عن مصادر البيانات المنفصلة باستخدام قواعد عامة.

ما هي لينكو؟What Is LINQ?.

لينكوLINQ، اختصار لـ: الاستعلام المتكامل للغة Language Integrated Query،وهي ليست مجرد تقنية واحدة ،ولكن تقريباً حوالي مليون تقنية دوت نت جديدة للفيجوال بيسك وجميعها تعمل بشكل ترادفي(بالتتالي) لجعل حياتك البرمجية أسهل.ليس أسهل في كل حالة.وكما مع أية تقنية تظهر للمرة الأولى،توجد محاسن ومساوئ.

المحاسن.The Good

تتواجد لينكو لأن بعض المبرمجين المصابين بالضجر لدى ميكروسوفت كانوا يحاولون الوصول للبيانات في قواعد بياناتهم بشكل مختلف عن ما يفعلوه بالملفات المعتمدة على البياناتfile-based data ،أو بياناتهم الكائنة في الذاكرة،أو بيانات XML.مع لينكو ،تتيح لك قاعدة مفردة الوصول إلى جميع هذه الصفات المميزة للبيانات وأكثر.القاعدة نفسها مشابهة لسكول،لغة استعلام قاعدة البيانات المعروفة مسبقاً لك،أو مرفقات البرمجة خاصتك.

تتضمن الفيجوال بيسك2008 دعم لينكو من أجل جداول قاعدة بيانات مخدم سكول SQL Server والكائنات(من لينكو إلى سكولLINQ to SQL)،مجموعات بيانات آدو دوت نتADO.NET (من لينكو إلى آدو دوت نت LINQ to ADO.NET ومن لينكو إلى مجموعة البياناتLINQ to DataSet)،تجمعات الكائنات في الذاكرة مثل المصفوفات أو التجمعات العامةGeneric collections (من لينكو إلى الكائناتLINQ to Objects)،و XML(من لينكو إلى XML: LINQ to XML).مباشرةً بعد التحرير الرسمي لفيجوال أستوديو2008،أطلقت ميكروسوفت إطار عمل كينونة الدوت نتADO.NET Entity Framework (لينكو إلى الكينوناتLINQ to Entities)،والتي توفر دعم لينكو محسن لمخدم سكولSQL Server، أوراكلOracle ، DB2،ومنصات قواعد البيانات الأخرى.هذه بداية عظيمة،ولكن الأخبار الجيدة لن تنتهي هناك.

إن لينكو قابلة للتمدد(التوسعextensible).وهذا يعني أنك تستطيع تحسين لينكو بحيث تستطيع الاستعلام عن أي نوع من البيانات تعينه.من لينكو إلى اللوحة الجدولية LINQ to Spreadsheet،من لينكو إلى ملف مؤشر (محدد)بمنظم جدوليLINQ to Tab-Delimited-File ،ومن لينكو إلى محتوى مقطع ديفيدي LINQ to DVD-Chapter-Content كل هذا متاح.بقدر الإثارة لكل هذه الإمكانيات،فليس لدي مساحة كافية في هذا الكتاب لأريك كيف تبرمجها،وكذلك توجد أخبار سيئة قادمة.

المساوئ. The Bad

لينكو نظام توسعي للاستعلام عن البيانات،فحالما تعمل على تأسيس اتصال بين عبارات الاستعلام والبيانات.فمن أجل بعض المواصفات المميزة للينكو،وخاصةً (من لينكو إلى كائنات)،لايوجد الكثير من الاتصال،لذلك فإن الاستعلام يكون خاطفاً.من أجل تنوعات لينكو الأخرى،وخاصة فرز قاعدة البيانات،عليك إنشاء فئات وسيطة والتي تربط متطلباتك بالبيانات.إن لينكو تقنية عمومية بإمكانها التفاعل مع أية بيانات حالما توفر المادة اللاصقة.وتلك المادة اللاصقة يمكن في بعض الأحيان أن تصبح لاصقة جداً.

وكمثال،خذ لينكو إلى سكول.تنفيذ لينكو هذا يحتاج إلى فئة تمثل الجداول والسجلات والتي ستستعلم عليها خلال لينكو.وهذه الفئات ليست صعبة الإنشاء،وتبدو مثل جداول قاعدة البيانات الأصلية.

مهما يكن،إذا عدلت تركيب جدولك،ستحتاج إلى تعديل الفئة الوسيطة لتنتفع من التغيرات الحاصلة على الجدول.إنها مهمة ستكون بحاجة لعملها على أية حال،حتى بدون لينكو،ولكن هذا شيء ما عليك أن تضعه في ذاكرتك .

طبيعة وسيط لينكو يعني أيضاً أن بعض معالجة البيانات يمكن أن تكون أبطأ عند مقارنة إتمام نفس المهمة بدون لينكو.الطبقات الإضافية من البيانات والكود يعني أشياء إضافية يجب على الكمبيوتر عملها.ولكن يقطن هذا مسبقاً في إطار عمل الدوت نت،لذلك لن أتجنب لينكو من أجل هذا السبب.

دعم التقنيات Supporting Technologies

إن لينكو صفقة كبيرة بالنسبة لميكروسوفت وإطار عمل الدوت نت.معظم الميزات الجديدة المضافة إلى الفيجوال بيسك2008 تم تقديمها بشكل رئيسي لدعم لينكو.قبل أن ندخل في استخدام لينكو لنلقي نظرة سريعة على التقنيات المضمنة في جعل لينكو ممكنة.

- تعابير الاستعلامQuery expressions،قلب(أو مركز)إمكانية الوصول للبيانات من خلال لينكو.يناقش هذا الفصل تعابير الاستعلام بالتفصيل.
- تعابير لمداLambda expressions،التي تم مناقشتها في الفصل 9.
- طرق التمديدExtension methods،التي تم تغطيتها في الفصل 12.
- استنتاج النوع المحليLocal type inference،التي تم مناقشتها في الفصل 6.
- الأنواع الغير مسماة Anonymous types،شيء ما جديد للفيجوال بيسك في2008،ولكن أيضاً شيء لم أناقشه حتى الآن.سأعطيك التفاصيل بعد الانتهاء من هذا المقطع.

.التفاويض الحرةRelaxed delegates،ميزة تتيح للفيجوال بيسك عمل تخمينات معتمدة على المعرفة والتجربة(ثاقبة) كما فيما إذا طريقة وتفويض يتطابقان أم لا.وهي مشابهة لاستنتاج(الاستدلال) على النوع،ولكن من أجل التفاويض ،استنتاج التفاويض بدل الأنواع البسيطة.

.حرفية XML Literals،الخاصيات المحورية لـ XML،تعابير XMLالمضمنة(المغلقة)،ودعم فضاء أسماء XMLضمن كودك المصدري،من المحتمل أنك تتذكر كل ما يخص هذه الميزات من الشرح في الفصل13.

.الأنواع"بدون قيمة Nullable "،التي تم مناقشتها في الفصل 6،مع بعض المناقشة الموسعة في فصل العمومياتGenerics ،الفصل16.

.الطرق الجزئيةPartial methods ،أول ما ظهرت معنا في الفصل 8.

.بادئات الكائنObject initializers ،موضحة في الفصل9.

.لغات أخرى وميزات مترجم compiler features جديدة ولن تكون بتلك الأهمية بما أنها لم تحصل على أسماء جديدة مضبوطة مذكورة خاصة بها.

الأنواع المجهولة الاسم.Anonymous Types

الأنواع المجهولةالاسم ميزة جديدة في الفيجوال بيسك لدعم لينكو،ولكن تستطيع استخدامها في كودك الخاص أيضاً.وهي بالضبط ما يفصح عنه اسمها:أنواع بدون أسماء. حسناً هذا ليس دقيق تماماً.الأنواع التي تملك أسماء،ولكن تم توليدها بشكل آلي من قبل مترجم الفيجوال بيسك،وهي لاتظهر بشكل مباشر في كودك المصدري.اعتبر فئة نموذجية تم تصميمها لحفظ معلومات على اختيارات السوشي sushi (طبق سمك ياباني).

```
Class Sushi
        Public FishName As String
        Public ServingCost As Decimal
    End Class
```

إنشاء حالة من هذه الفئة بسيط جداً.

```
Dim tastyFood As New Sushi
        tastyFood.FishName = "maguro"
        tastyFood.ServingCost = 3.5@
```

أو ،باستخدام قاعدة مُسنِد الكائن *object initializer* التي تحدثت عنها في الفصل 9،تستطيع إنشاء حالة وملئ حقولها،الكل في عبارة واحدة.

```
Dim tastyFood As New Sushi With {.FishName = "maguro", .ServingCost = 3.5@}
```

تأخذ الأنواع المجهولة الاسم هذا التركيب المصقول مع خطوة أبعد من ذلك وذلك بشطب اسم الفئة تماماً.

```
Dim tastyFood = New With {.FishName = "maguro", .ServingCost = 3.5@}
```

الحالة tastyFoodهي الآن حالة من فئة ما مع عضوين،نصي مسمىFishName ،وقيمة عشرية مسماة ServingCost.الشيء الوحيد الذي ليس لديها هو اسم الفئة وهو معروف بالنسبة لك.ولكن الفيجوال بيسك تعرف ما هو.من أجل الترويج عن النفس قمت بترجمة ذلك المقطع الأخير من الكود وبحثت عن الاسم للنوع المولد .وهو التالي:

```
VB$AnonymousType_0`2<T0,T1>
```

ما هو مهم حقاً هو أن الفيجوال بيسك عملت على إنشاء نوع عمومي(شمولي) مع نوعي وسيط لحافظة (حاجز مكانtype parameter placeholders): T0 (من المحتمل أنه متصل بالعضو النصي FishName) و `T1` (من المحتمل أنه على صلة بـ ServingCost العشري).

الأنواع المجهولة الاسم هي مستخدمات رئيسية لاستنتاج النوع.تخمن الفيجوال بيسك نوع البيانات لكل عضو بالاعتماد على البيانات التي تزودها مع كل اسم. في حالات sushi، العضو ServingCostمن نوع العشري Decimalبالاعتماد على محرف العشري `@` المزود مع تعريف الحالة.

لينكو إلى كائنات.LINQ to Objects

تتيح لك لينكو الاستعلام عن البيانات من عدة مصادر مختلفة للبيانات،وكل تفاعل لينكو على بيانات LINQ-to -data interaction يتم إدارته بواسطة موفر لينكو *LINQ provider*.لقد عملت على جدولة الموفرات المضمنة في الفيجوال بيسك2008 بعد قليل،فجميعها لديها الاسم"لينكو على أي شيءLINQ to *something*" بالنسبة لي الموفرات الأبسط هي لينكو على كائنات، المصممة للتفاعل مع مجموعات الكائنات في الذاكرة. تتيح لك لينكو على كائنات معالجة استعلامات معتمدة على تجمعات كائن object collections،مصفوفات فيجوال بيسك،وأي كائن يدعم واجهات الدوت نت IEnumerable أوIEnumerable(Of T)، من ضمنها التجمعات الخاصة بك.(الكائنات المتنوعة ضمن عالم آدو دوت نت ADO.NET تدعم هذه الواجهات،ولكن هذه الأنواع تقع تحت موفر لينكو على مجموعة بياناتLINQ to DataSet ،سيتم مناقشتها بعد قليل).عندما تشغل لينكو على استعلامات كائناتLINQ to Objects queries ، فمخرجات الاستعلام هي مجموعة جديدة من الكائنات التي تحتوي مجموعة جزئية من بيانات كائن المصدر الأصلي.يتيح لك هذا تشغيل استعلامات بقول الأشياء كـ"مرحباً لينكو،أعطني أسماء هؤلاء الموظفين وخصوصياتهم فقط،من هذه القائمة للموظفين وخصوصياتهم ،والذين تم تعينهم في التسعين يوم الماضية".هذا يعمل على إنتاج مجموعة، أي تجمع معتمد على IEnumerable،يمكن أن يتم الاستعلام عنه أبعد من ذلك أو يتم استخدامه كلما احتجت لأي تجمع أخر في كود الفيجوال بيسك.

ملاحظة.

على الرغم من أن لينكو لديها عدد محدد من المعاملات والكلمات المحجوزة،فيمكن أن يتم استخدامها في تشكيلة غنية من الجمع(الضم)،وسيتم تقديم بعضها فقط في هذا الفصل.من أجل أمثلة و شروحات أكثر للقواعد،راجع مقطع لينكو LINQلتوثيق ميكروسوفت MSDN المضمن مع نسخة الفيجوال أستوديو.

قبل الدخول في موفرات لينكو الأكثر تعقيداً،دعنا نستكشف قواعد استعلامات لينكو باستخدام "لينكو على كائنusing LINQ to Objects ".في المقاطع القليلة القادمة،سأستخدم تجمعين صغيرين في الذاكرة من الكتب كبيانات مصدرية لاستعلامي.إليك تعريف الفئة من أجل كل كتاب والذي يتضمن القليل من الأعضاء المقبولة.

```
Class Book
    Public Title As String
    Public AuthorID As String
    Public Pages As Integer
End Class
```

يظهر المؤلفين Authors في فئة.حالات الكتب والمؤلفين تتطابق من خلال الحقل المشترك AuthorID .

```
Class Author
    Public AuthorID As String
    Public FullName As String
End Class
```

سأعمل على إنشاء تجمعات صغيرة لإدارة المؤلفين authorsوالكتبbooks .

```
Dim Writers As New Generic.List(Of Author)
Dim Library As New Generic.List(Of Book)
        Writers.Add(New Author With {.AuthorID = "LT", .FullName = "Tolstoy, Leo"})
        Writers.Add(New Author With {.AuthorID = "LW", .FullName = "Wallace, Lew"})
        Writers.Add(New Author With {.AuthorID = "JB", .FullName = "Barrie, J. M."})
        Library.Add(New Book With   {.Title = "War and Peace",.AuthorID = "LT", .Pages = 1424})
        Library.Add(New Book With   {.Title = "Anna Karenina", .AuthorID = "LT", .Pages = 976})
        Library.Add(New Book With   {.Title = "Ben-Hur", .AuthorID = "LW", .Pages = 544})
        Library.Add(New Book With   {.Title = "Peter Pan",.AuthorID = "JB", .Pages = 192})
```

لجعل فهمنا أسهل لمخرجات كل استعلام،دعنا نتظاهر أننا قد كتبنا طريقة تعرض نتائج أي استعلام في نموذج جدول.سأستدعي الروتين ShowResults .

تعابير الاستعلام الأساسية.Basic Query Expressions

يتم بناء تعابير لينكو من شروط استعلام*query clauses* لديها نفس الصفة الخاصة كما في شروط عبارات سكول على مستوى قاعدة البيانات. مع استثناء شرط From،والذي يجب أن يظهر أولاً، الشروط الأخرى يمكن أن تظهر في أي ترتيب ضمن الاستعلام بشكل عام.

الشرط "من" .The From Clause

كل استعلام لينكو LINQأساسي يبدأ بالكلمة المحجوزة "من From ".

```
Dim bookBag = From bk In Library
        ShowResults(bookBag)
        ' Results --> War and Peace LT 1424
        ' Anna Karenina LT 976
        ' Ben-Hur LW 544
        ' Peter Pan JB 192
```

هذا الاستعلام المكون من أربع كلمات إلى حدما أقصر استعلام لينكو تستطيع كتابته.عملت على تخزين نتائج الاستعلام في المتغير bookBag(حيث يتم الاستدلال على نوع بياناته بواسطة الاستعلام)،ولكن يمكن أن يتم استخدام الاستعلام مباشرةً كتعبير أيضاً.

```
ShowResults(From bk In Library)
```

المتغير bk المضمن في الاستعلام يُعرَف بمتغير مجال *range variable* أو متغير دوران *iteration variable* (ليس عليك استخدام bk ،فقد اخترت هذا الاسم بشكل عشوائي،فهو متغير لذلك امنحه الاسم الذي تريد) يوفر هذا المتغير طريقة لتحديد كائنات وأعضاءها من مصدر بيانات ضمن withinالاستعلام.بما أن Libraryهو تجمع،فليس من المعقول القول Library.Title عند الإشارة إلى عنوان كتاب فقط،بالمقابل تشير إليه بـ bk.Title.

شخصياً،إني أرى هذه القاعدة *source* In *variable* نوعاً ما غير مباشرة .والأفضل أكثر قاعدة الاسم المرادف –للجدول table-aliasالمستخدمة في استعلامات سكول.

```
SELECT * FROM Library AS bk
```

تنجز كلمة سكول المحجوزة ASإلى حدٍ بعيد نفس وظيفة كلمة لينكو المحجوزةIn .بالرغم من قلقي الداخلي،فإن القاعدة In هي السائدة،فلن تستطيع استخدام التركيب ASفي لينكو بما أن الكلمة المحجوزة ASفي الفيجوال بيسك يتم استخدامها لإسناد نوع البيانات.

The Select Clause."اختر" الشرط

إذا استخدمت الشرط "من" From " فقط في استعلامك،فإنه يعود بجميع البيانات من مجموعة الكائن الأصلي،متضمناً جميع الأعضاء.إذا كنت تريد تحديد النتائج بحيث يتم تضمين بعض الأعضاء فقط،استخدم الشرط Select لتحديد الحقول التي يجب أن يتم تضمينها.

```
Dim bookBag = From bk In Library Select bk.AuthorID, bk.Title
        ShowResults(bookBag)
        ' Results --> LT War and Peace
        ' LT Anna Karenina
        ' LW Ben-Hur
        ' JB Peter Pan
```

مجموعة النتائج لهذا الاستعلام الجديد يحذف عدد الصفحات الموجودة في البيانات الأصلية.وهذا لأن استعلام لينكو طلب فقط حقل معرف المؤلف AuthorIDوحقل العنوانTitle ، ولم يتم عمل استعلام على العضو Pagesمن خلال الشرط Select.لاحظ أيضاً أني عملت على عكس ترتيب كل من حقول معرف المؤلف والعنوان من تعريف الفئة الأصلية.وهذا القلب تم عكسه في النتائج المطبوعة أسفل الكود السابق باللون الأخضر.

ماوراء الكواليس،تعمل لينكو على إنشاء نوع جديد مجهول الاسم new anonymous type والذي يتضمن عضوين: حقل AuthorIDالنصي وحقل Titleالنصي.حالة واحدة لهذا النوع المجهول الاسم يتم إنشائها من أجل كل سجل استعلام ناتجresultant query record .ومن ثم يتم تحزيم bundled up هذه الحالات في تجمع جديد معتمد على IEnumerable(Of T).يتيح لك هذا استخدام نتائج الاستعلام في استعلام جديد،أو في أي كود سيتفاعل مع تجمع من النتائج بشكل طبيعي،مثل العبارة For Each.

```
Dim bookBag = From bk In Library Select bk.AuthorID, bk.Title
        For Each oneResult In bookBag
            MsgBox(oneResult.Title)
        Next oneResult
        ' The Loop Displays --> War and Peace
        ' Anna Karenina
        ' Ben-Hur
        ' Peter Pan
```

بالإضافة إلى ترحيل الحقول من الكائنات الأصلية إلى مجموعة النتائج،تستطيع استخدام المعاملات والدوال لتعديل النتائج.فالمثال التالي يستخدم الدالة StrReverseلتبديل اسم العنوان قبل ترجمة النتائج.

```
Dim backward = From bk In Library Select StrReverse(bk.Title)
        ShowResults(backward)
        ' Results --> ecaeP dna raW
        ' aninerаK annA
        ' ruH-neB
        ' naP reteP
```

ملاحظة:

على الرغم من أننا مازلنا إلى حدما في بداية مناقشتنا للينكو،عليك أن تعلم الآن أن العمل مع لينكو يتطلب الكثير من التجريبexperimentation .بالرغم من أن هدفه الثبات،فإن لينكو ملئ بالمفاجآت. على سبيل المثال،لايعمل المثال السابق على إنشاء مجمع نوع مجهول الاسم الذي توقعته.بالمقابل فهو يميز مجموعة النتائج المحتوية على نصوص(أو سلاسل حرفية)فقط،وينشئ مجموعة نصية بسيطة بدلاً من مجمع أنواع مع عضو نص.كن متنبه ضد صدمات قليلة مثل هذه عند كتابة استعلامات لينكو.

The Distinct Clause."متميز" الشرط

بشكل افتراضي،يعمل الشرط Selectعلى إعادة جميع السجلات من المصدر.فالحصول على معلومات كاملة شيء جيد،ولكن في بعض الأحيان إن الكثير من الأشياء الجيدة، تحتوي تكرارات .على سبيل المثال، يعمل هذا الاستعلام على إعادة معرف المؤلفين فقط من أجل كل كتاب متاح.

```
Dim justIDs = From bk In Library Select bk.AuthorID
        ShowResults(justIDs)
        ' Results --> LT
        ' LT
        ' LW
        ' JB
```

النتائج كاملة ،ولكن ظهر LT مرتين.بالاعتماد على حاجاتك من المحتمل أن يكون ذلك شيء سيء.بإضافة الشرط Distinct،تستطيع التخلص من(أو استئصال weed out) التكرار الغير مرغوب.

```
Dim justIDs = From bk In Library Select bk.AuthorID Distinct
        ShowResults(justIDs)
        ' Results --> LT
        ' LW
        ' JB
```

تنظر الكلمة المحجوزة Distinctعلى كامل السجلات من أجل التكرارات.يتم إخراج السجل فقط إذا كانت جميع الحقول في ذلك السجل تطابق تماماً جميع الحقول في سجل آخر.

The Where Clause. "حيث"الشرط

بينما يتيح لك الشرط استئصال weed out الحقول الغير مرغوبة،يتيح لك الشرط استبعاد eliminateكائنات كاملة بالاعتماد على معيار criteria تعينه أنت.

```
Dim bigBooks = From bk In Library Where bk.Pages >= 1000
        ShowResults(bigBooks)
        ' Results --> War and Peace LT 1424
```

يختبر هذا الاستعلام جميع سجلات المصدر القادمة في تجمع المكتبة Libraryويعمل على تضمين كائنات المصدر في النتائج إذا كانت فقط لديها عدد صفحات 1،000 أو أكثر. يمكن لشرط أن يصبح معقداً،مع ضم معايير متعددة من خلال الكلمات المحجوزة OrgAnd، وتجميعها ضمن أقواس.

```
  Dim choices = From bk In Library Where bk.Pages >= 1000 Or (bk.Pages < 1000 And InStr(bk.Title, "-") > 0) Select bk.Title
          ShowResults(bigBooks)
' Results --> War and Peace
        ' Ben-Hur
```

يبين هذا الاستعلام كيفية تضمين ميزات لا تمت إلى لينكو بصلة مثل الدالة InStr، في المعايير،مما يسمح لك بتقييد النتائج بالاعتماد على نتائج محسوبة.

الشرط "تريب بـ" The Order By Clause.

من غير المضمون ظهور نتائج لينكو ،المعتمدة على مصدر بيانات،في أي ترتيب معين .لإنتاج نتائج استعلام في ترتيب معين،استخدم الشرط"ترتيب بواسطةOrder By ".الكلمات المحجوزة Order By تسبق واحد أو أكثر من حقول المصدر أو القيم المحسوبة،أو المحددة بواسطة الفاصلة،وتستطيع بشكل اختياري تضمين الكلمة المحجوزة تصاعدي Ascendingأو تنازلي Descendingلعكس ترتيب الفرز لكل حقل مفروز.(الفرز التنازلي هو الافتراضي لكل حقل)

```
Dim bookBag = From bk In library Select bk.pages, bk.title Order By pages Descending
        ShowResults(bookBag)
        ' Results --> 1424 War and Peace
        ' 976 Anna Karenina
        ' 544 Ben-Hur
        ' 192 Peter Pan
```

ملاحظة.

إذا كنت لا تريد إظهار النتائج بواسطة الروتين ShowResults،اعمل على إنشاء الفئة التالية:

```
Class book
    Public title As String
    Public authorId As String
    Public pages As Integer
End Class
```

أضف إلى الفورم صندوقListBox1 قائمة ومن ثم في معالج حدث تحميل الفورم اعمل على إضافة الكود التالي:

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Dim library As New System.Collections.Generic.List(Of book)
        library.Add(New book With {.title = "bk1", .authorId = "mmm", .pages = 1200})
        library.Add(New book With {.title = "bk2", .authorId = "nmm", .pages = 230})
        library.Add(New book With {.title = "bk3", .authorId = "mnm", .pages = 2300})
        library.Add(New book With {.title = "bk4", .authorId = "mmn", .pages = 200})
        Dim bookBag = From bk In library Select bk.pages, bk.title Order By pages Descending
                For Each bk In bookBag
            ListBox1.Items.Add(bk.pages & vbTab & vbTab & bk.title)
        Next
```

ومن ثم شغل التطبيق ولاحظ النتائج.

الحقول المضمنة في الشرط Order By يجب أن يتم إحضارها في الشرط Select،مع التخلي عن سابقة متغير المجال(أي في هذه الحالة لا تكتب `( Order By bk.pages …`

إذا استخدمت الشرط Fromدون الشرطSelect، عليك تضمين سابقة متغير المجال في حقول Order By.

ضم المصادر.Joining Sources

إذا كنت تريد فقط الاستعلام عن البيانات من تجمع بيانات أو مصدر مفرد،من المحتمل أن لاتحتاج إلى شيء مثل لينكو في المكان الأول.عندما يحين الوقت لدمج النتائج من جداول مختلفة،يوفر مرةً أخرى لينكو تركيب مشابه لسكول لضم الجداول.عملياً،إنه يوفر تشكيلتين،توازيان تشكيلات التركيب المدعومة من قبل بائعي سكول.

القاعدة الأولى تستخدم الكلمة المحجوزة "ضم Join "لتعين وصل حقل معين.الاستعلام التالي "يضم داخلياًinner joins " الجدول Libraryوالجدول Writersعند نقطة الاتصال AuthorIDالمتوقعة.

```
im bookBag = From bk In Library Join au In Writers On bk.AuthorID Equals au.AuthorID Select bk.Title, au.FullName Order By bk.Title
        showresults(bookBag)
        ' Results --> Anna Karenina Tolstoy, Leo
        ' Ben-Hur Wallace, Lew
        ' Peter Pan Barrie, J. M.
        ' War and Peace Tolstoy, Leo
```

الكلمات المحجوزة الخاصة Onو Equalsتساعد في إتمام تركيب الربط.إذا تضمن الربط خاصتك عدة مفاتيح،تستطيع استخدام الكلمة المحجوزة Andلتخصيص ربط المفاتيح المختلفة.

```
Dim results = From t1 In Table1 Join t2 In Table2 On t1.Key1 Equals t2.Key1 And t1.Key2 Equals t2.Key2
```

تركيب الربط أو الضم الثاني يتيح لك استخدام الشرط Whereللدلالة على روابط الحقول.

```
Dim bookBag = From bk In Library, au In Writers Where bk.AuthorID = au.AuthorID Select bk.Title,u.FullName Order By bk.Title
'             نفس نتيجة الكود السابق
```

تتضمن لينكو تشكيلة أخرى للربط تولد نتائج استعلام هرمية التنظيم*hierarchical query results*. في مثل هذه الاستعلامات،واحد من الحقول في كل سجل ناتج سيكون تجمع يحتوي نتائج متعددة.ويسمح هذا التركيب للينكو بالعمل على إعادة قائمة بجميع المؤلفين،مؤلف في كل صف row، حيث أن كل سجل مؤلف يتضمن حقل"الكتبbooks "،ومن المحتمل مع قيم متعددة.

```
Dim authorBooks = From au In Writers Group Join bk In Library On au.AuthorID Equals bk.AuthorID Into Published = Group Select au.FullName, Published Order By FullName
        ShowResults(authorBooks)
```

```
        ' Results --> Barrie, J. M. Peter Pan
        ' Tolstoy, Leo War and Peace
        ' Anna Karenina
        '  Wallace, Lew Ben-Hur
```

هذا الاستعلام لديه نوعاً ما تركيب غريب،ولكن يعمل بنجاح على إنشاء مجموعة نتائج مع عمودين:الاسم FullName(من أجل اسم المؤلف) والمنشور Published(من أجل تجمع من الكتب المنشورة من قبل مؤلف معين).من أجل كل سجل معاد،العضو Published هو تجمع تابع يمكن أن يتم معالجته مثل التجمعات الأخرى.

التخطي والأخذ .Skip and Take

يتيح لك الشرط تخطي Skipالسجلات الأولى بعدد xفي مجموعة النتائج،وبصفة مؤثرة إلقائهم مع النفايات،مثل قشور الموز.يعمل الشرط Takeالعكس تماماً،يحفظ فقط أول عدة سجلات في النتائج المولدة. الاستعلام التالي يتخطى السجلين الأولين في تجمع البيانات الأصلي،مرجعاً فقط تلك السجلات التي تتبع القيم التي تم تجاهلها:

```
Dim someBooks = From bk In Library Select bk.AuthorID, bk.Title Skip 2
        ' Results --> LW Ben-Hur
        ' JB Peter Pan
```

تتيح لك الشروط SkipWhile وTake While ذات الصلة استخدام تعبير منطقي بدل عدد ما للدلالة على متى سيستمر تخطي أو أخذ سجلات.إن كل من Skip وTakeمفيدة من أجل تصفح النتائج،عند عرض صفحة واحدة فقط من النتائج في كل مرة من مجموعة أكبر من النتائج المستعملة.فمنطق مشابه للتالي يمكن أن يتم استخدامه لإظهار السجلات المقررة للصفحة الحالية CurrentPage فقط

```
Dim onePageWorth = From bk In Library Select bk.AuthorID, bk.Title Skip ItemsPerPage * CurrentPage Take
ItemsPerPage
```

كلمة تحذيرية بخصوص TakeوSkip :إنها تعمل على صنع اختلاف عندما تضعها في استعلامك.(سأشرح السبب التقني لهذا في مقطع التنفيذ المؤجل،فيما بعد في هذا الفصل)على سبيل المثال،خذ هذا الاستعلام المعتمد على بيانات كتاب الأصلية

```
Dim someBooks = From bk In Library Order By bk.Title Take 2
```

هذا الاستعلام يعمل على إعادة Anna Karenina متبوعاً بـ Ben-Hur،كما ستتوقع،ولكن إذا نقلت شرط خطوة،ستحصل على نتيجة مختلفة:

```
Dim someBooks = From bk In Library Take 2 Order By bk.Title
```

هذه المرة ،سيعمل الاستعلام على إعادة Anna Karenina متبوع بـ War and Peace.في الاستعلام الأول،تم تخزين محتوى بواسطة العنوان قبل أن يتم أخذ السجلين.في الاستعلام الثاني تم أخذ السجلين أولاً،وقبل أن يتم تطبيق أية عملية فرز عليهم.

لايتأثر فقط Take وSkipبهذا الترتيب.جميع الشروط في استعلامك تتأثر.فالتفكير بمنطق استعلامك شيء أساسي،بما أنه يمكن للشرط الموضوع في غير مكانه أن يعطيك نتائج غير متوقعة.

تحويل النتائج إلى تنسيقات أخرى.Converting Results to Other Forms

بما أن أي نتائج لأي استعلام لينكو تتوافق مع الواجهةIEnumerable(Of T)، فهي جاهزة لأن يتم استخدامها مباشرةً في استعلامات أخرى أو في ماسحات عددية enumerable scans. إذا كنت بحاجة الوصول إلى السجلات بطريقة تقليدية من أجل أهداف أخرى،توفر لينكو العديد من الميزات التي تعمل على نقل النتائج بسرعة ضمن مصفوفة arrayأو تجمع عمومي generic collection .

تتضمن كل نتيجة استعلام ثلاث طرق تعمل على إنجاز هذه التحويلات: إلى مصفوفة ToArray،إلى قاموسToDictionary ، وإلى قائمةToList. تحول ToArrayالنتائج إلى مصفوفة فيجوال بيسك قياسية،مع سجل نتيجة واحد يتم تخزينه في كل عنصر مصفوفة.

```
Dim queryResults = From ...
        Dim arrayVersion = queryResults.ToArray()
```

يقوم ToListبنفس العملية،إنشاء تجمع Generic.Listجديد بالاعتماد على نتائج الاستعلام.يعمل ToDictionary على إنشاء تجمع Generic.Dictionary،ولكن يجب عليك توفير دالة ToDictionaryتستخرج المفتاح.في معظم الحالات،تعبير لامدا الذي يحدد حقل المفتاح key field الذي سيفي بالغرض.

```
Dim authors = From au In Writers Order By au.FullName
        Dim authorDict = authors.ToDictionary(Function(x) x.AuthorID)
        ListBox1.Items.Add(authorDict("LW").FullName)
MsgBox(authorDict("LW").FullName)
        ' Results --> Wallace, Lew
```

استعلامات الحاصل(الإجمالي).Aggregate Queries

تتيح لك استعلامات لإجمالي"تحصيل sum up المعلومات من استعلام أكبر ضمن نتيجة مفردة أو مختصرة(مكثفةcondensed).بدل البدء بالكلمة المحجوزة From، فاستعلامات الحاصل النقية تبدأ بـالكلمة المحجوزةAggregate. كل استعلام إجمالي يستخدم واحد أو أكثر من دوال التجميع(أو الإجمالي,*aggregate functions*)،مثل الدالة Sumفي الاستعلام التالي:

```
Dim numBooks = Aggregate bk In Library Into Sum(bk.Pages)
        MsgBox(numBooks) ' Displays: 3136
```

تتضمن لينكو ثماني دوال إجمالي قياسية،مبينة في الجدول التالي.تقبل كل دالة تعبير يشير إلى ما يجب أن يتم تجميعه خلال الاستعلام.

الدالة Function	الشرح Description
All	تعود بقيمة منطقية تشير فيما إذا التعبير الممرر لها صواب أو خطأ بالنسبة لجميع السجلات.الشرط All(bk.Pages > 1000)سيعمل على إعادة خطأ بما أن كتاب واحد فقط لديه صفحات أكثر من 1000صفحة.
Any	مشابهة للسابقة،ولكن تعود بصواب إذا كان واحد فقط من السجلات يتطابق مع تعبير المعيار المزودة به.
Average	تعود بالمتوسط لأي تعبير تم تمريره لها.
Count	تعود بإحصاء السجلات مع نتائج تعبير صواب True .للعودة بعدد جميع السجلات في استعلام استخدم Count(True) .
LongCount	نفس Count،ولكن تعود بلونغ(طويل Long)عوضاً عن انتغر Integer .
Max	تعود بالتعبير العددي الأكبر من مجموعة سجلات.
Min	تعود بالتعبير العددي الأصغر من مجموعة سجلات.
Sum	تعود بمجموع تعابير عددية من مجموعة سجلات.

لو عملت على تضمين أكثر من دالة إجمالي في الاستعلام،فإن مجموعة النتائج هي سجل وحيد يتضمن عدة حقول مسماة(مميزة بالاسم).استخدم الاسم المستعار aliasقبل دالة الإجمالي لمنحها اسم.(الأسماء المستعارة متاحة في جميع أنواع الاستعلام،وليس فقط في الإجمالي).

```
Dim numBooks = Aggregate bk In Library Into TotalPages = Sum(bk.Pages), AvgPages = Average(bk.Pages)
        MsgBox(numBooks.AvgPages) ' Displays: 784
```

تستطيع أيضاً تضمين تعابير إجمالي في استعلام قياسي لايمت إلى الإجمالي بصلة.يعود الاستعلام التالي بعدد الكتب المكتوبة بواسطة مؤلف،باستخدام دالة الإجمالي"إحصاء count" لجمع نتائج كل مؤلف:

```
Dim authorBooks = From au In Writers Group Join bk In Library On au.AuthorID Equals bk.AuthorID Into
NumBooks = Count(True) Select au.FullName, NumBooks Order By FullName
        'ShowResults(authorBooks)
        ' Results --> Barrie, J. M. 1
        ' Tolstoy, Leo 2
        ' Wallace, Lew 1
        For Each au In authorBooks
            ListBox1.Items.Add(au.FullName & vbTab & au.NumBooks)
        Next
```

Advanced Query Expressions.تعابير الاستعلام المتقدمة

من المحتمل أنك تتذكر من الفصل9 أن تعابير لامدا يتم تحديثها من قبل المترجم ضمن شيء أبسط. في حال فشل استعلاماتك في جزيئات أقل من حجم الذرة،فإن المعالج CPU جاهز ليتصرف.ولكن ليس عليك البدء بالاستعلامات الكاملة.تستطيع إنشاء استعلامات خاصة بك باستخدام طرق التوسيع extensionوتعابير لامداlambda. يتم إرفاق الطرق الموسعة التي نتكلم عنها إلى واجهة IEnumerable.وهذا يعني أن أي شيء يبدو مشابه لتجمع أو مصفوفة يمكن أن يتم تضمينه في استعلامات معتمدة على طرق التوسيع، باستخدام تعابير لامدا كمعاملات نسبية.

لنعمل على تحويل واحد من استعلاماتنا السابقة إلى طريقة التوسيع extension method cالنظيرة.

```
Dim bigBooks = From bk In Library Where bk.Pages >= 1000
```

إنه الاستعلام الذي يعود بالكتب الكبيرة فقط .نفس الاستعلام الذي يستخدم طرق التوسيع يبدو مشابه للتالي:

```
Dim bigBooks = Library.Where(Function(bk) bk.Pages >= 1000)
```

في هذا المثال الطريقة"Where "هي بالفعل طريقة توسيع لواجهةIEnumerable، والتي تتضمن أيضاً Select، OrderBy، Join، GroupJoin،Count ، Max،Min وطرق أخرى تتوافق مع المعاملات ضمن لغة استعلام لينكو.وكما شرحت في الفصل12،بإمكانك إضافة طرق التوسيع الخاصة بك إلى الواجهةIEnumerable ، مما يمنحك حتى طرق أكثر لتخصيص استعلامات لينكو التي تقوم بها.

لينكو على XML اكس ام إل. LINQ to XML

قدمت في الفصل13 حرفيات *XML* ، محتوى *XML* الذي يتم تضمينه في الكود المصدري للفيجوال بيسك تماماً.عندما تضع لينكو في الصورة،فسيكون لديك فجأةً طريقة لتوليد مستندات *XML*ضخمة وذلك بضم مجموعة من السجلات مع قالب الحرفية لـ *XML*.

المقطع التالي من الكود يعمل على إنشاء مستند XMLباستخدام كل من تجمع Libraryوتجمع Writersالذي عملناهما سابقاً في بداية الفصل،وذلك بممازجة لينكو و XML بطريقة تعمل صداع في راسي.

```
Dim bookXML As XDocument = _
        <?xml version="1.0"?>
        <booklist>
            <%= From bk In Library Join au In Writers On bk.AuthorID Equals au.AuthorID Order By
bk.Title Select _
                <book>
                    <title><%= bk.Title %></title>
                    <author><%= au.FullName %></author>
                    <pages><%= bk.Pages %></pages>
                </book> %>
        </booklist>
        bookXML.Save("books.xml")
```

لاحظ كيفية توجب وضع رموز استمرارية السطر في الكود الخاص بلينكو،ولكن ليس في حصة XML.في الواقع إني أكره هذا،ولكن ألم يعمل على إنتاج XML جميل.إذا ألقيت نظرة على ملف *books.xml*الناتج عن هذا الكود،فإنه يحتوي محتوى مدمج من XML و تجمعاتنا الأصلية.ولقد تم ترك فراغات بشكل جميل.

```
<?xml version="1.0" encoding="utf-8" ?>
- <booklist>
- <book>
  <title>Anna Karenina</title>
  <author>Tolstoy, Leo</author>
  <pages>976</pages>
    </book>
- <book>
  <title>Ben-Hur</title>
  <author>Wallace, Lew</author>
  <pages>544</pages>
    </book>
- <book>
  <title>Peter Pan</title>
  <author>Barrie, J. M.</author>
  <pages>192</pages>
    </book>
- <book>
  <title>War and Peace</title>
  <author>Tolstoy, Leo</author>
```

```
  <pages>1424</pages>
    </book>
    </booklist>
```

مفتاح تمازج XMLولينكو هو وضع المحددات =%>و<% حول كود تعيين لينكو.إذا ألقيت نظرة إلى العينة بحذر،سترى أن هناك مجموعتين من المحددات، واحدة داخل الأخرى.

```
<%= From ...
<title><%= bk.Title %></title>
... %>
```

المجموعة الخارجية من المحددات تحيط كامل استعلام لينكو،بينما كل مجموعة داخلية من المحددات تعين متغير استبدال للتضمين في محتوى XML.
بقدر سهولة إنتاج XML باستخدام لينكو،فهو سهل تماماً لاستعلام البيانات من مستندات XML الموجودة.لإعادة تحميل XMLعملنا فقط على حفظه بما يسمح لنا الاستعلام عن قائمة عناوين الكتب بتمازج لينكو مع خاصيات XMLالمحورية.

```
Dim bookXML As XDocument = XDocument.Load("books.xml")
        Dim fromXML = From bx In bookXML...<book> Select bx.<title>.Value
        ' ShowResults(fromXML)
        ' Results --> Anna Karenina
        ' Ben-Hur
        ' Peter Pan
        ' War and Peace
        For Each bk In fromXML
            ListBox1.Items.Add(bk)
        Next
```

لينكو من أجل آدو دوت نت ـ فيما يتعلق بالبيانات.LINQ for ADO.NET-Related Data

إن كل من آدو دوت نت ولينكو يعملان الآن مع بعضهما بشكل جيد.في الحقيقة،يدعم آدو دوت نت ثلاثة موفرات لينكو.

لينكو على كينونات.LINQ to Entities

حالما أطلقت ميكروسوفت فيجوال أستوديو2008،أطلقت ميكروسوفت إطار عمل كينونة آدو دوت نتADO.NET Entity Framework. هذه الوسيطة(الواجهة) interface بين كودك البرمجي وقاعدة البيانات سيتيح لك تعريف عرض منطقي *logical*لنظامك.على سبيل المثال،تستطيع إنشاء كينونة entityتدعى "الجداولtables "،الكل في عرض منطقي.ويمكن أيضاً أن تكون الإجراءات المخزنة stored procedures ذات الصلة جزء من الحزمة. يقوم إطار العمل بعمل كل هذا السحر بإنشاء مجموعة وسيطة من الفئات ومحتوى XMLالتركيبي ذو الصلة (المناسب) الذي يدير الربط بين المنطق والعروض الجسدية للبيانات.هذه الفئات يمكن أن يتم استخدامها فيما بعد في استعلامات لينكو،ومؤلفي الاستعلامات لايحتاجون إلى القلق بخصوص الأمور العادية مثل اتصالات قاعدة البيانات ومراجع المفاتيح الأجنبية(الثانويةforeign key references). في الحقيقة،إن المبرمجين قد كتبوا كود مثل هذا منذ سنوات،عاملين على تجريد(تلخيص) نمط البيانات الجسدي في عرض منطقي وهو أسهل للبرمجة عليه.يجعل إطار عمل الكينونة ببساطة هذه المعالجة أسرع وأسهل التثبيت.
يتضمن إطار العمل عدة أدوات تساعدك في بناء الكينونات من تراكيب قاعدة البيانات المصدرية.واحدة من الأدوات الرئيسية هي مصمم نمط بيانات كينونة آدو دوت نت ADO.NET Entity Data Model Designer، أداة السحب والإسقاط المرئية visual drag-and-drop tool التي تجعل إنشاء كينونات entitiesبقدر سهولة إنشاء نماذج الفيجوال بيسك.
ولأن إطار عمل كينونة آدو دوت نت تقع خارج الفيجوال أستوديو2008،فلن أعمل على شرح إطار العمل في هذا الكتاب.

لينكو على مجموعة بيانات.LINQ to DataSet

تدعم لينكو استعلامات السجلات ضمن جداول بيانات آدو دوت نت.لاتدعم كائنات جداول بيانات DataTableآدو دوت نت بشكل مباشر الواجهة IEnumerable،فالحقول ضمن هذه الجداول وبشكل افتراضي غير محددة النوعuntyped ،مما يجعل لينكو يغضب.يتغلب التخصص الوظيفي "لينكو على مجموعة بيانات *LINQ to DataSet* " على كل من هذين القصورين limitations لذلك فإن استعلام مجموعة البيانات data sets يعمل.
فيما سبق في هذا الفصل ،شاهدنا أمثلة لينكو التي استخدمت فئة Book.دعنا نحفظ عينة البيانات تلك،ولكن مع التظاهر أن البيانات تظهر الآن في نسخة جدول بيانات DataTable آدو دوت نت .سيكون للجدول أربع سجلات (من أجل الكتب في مثالنا)وثلاث أعمدة: العنوانTitle، معرف المؤلفAuthorID، والصفحاتPages .

```
Class Book
    Public Title As String
    Public AuthorID As String
    Public Pages As Integer
End Class
```

فبدلاً من إصدار استعلام لينكو على كائنات مثل التالي:

```
Dim choices = From bk In Library Where bk.Field(Of Integer)!Pages >= 1000 Or (bk.Pages < 1000 And
InStr(bk.Title, "-") > 0) Select bk.Title
```

يستخدم لينكو على مجموعة بيانات طرق تعين كائن مجموعة البيانات التي تجبر كائنات آدو دوت نت ضمن شيء ما حيث تستطيع لينكو التفاعل معه،وفي نموذج محدد النوع بقوة.

```
Dim choices = From bk In bookTable.AsEnumerable() Where bk.Field(Of Integer)("Pages") >= 1000 Or
(bk.Field(Of Integer)("Pages") < 1000 And InStr(bk.Field(Of String)("Title"), "-") > 0) Select New With
{.Title = bk.Field(Of String)("Title")}
```

إنها تبدو بطريقة مختلفة حقاً،ولكنه نفس الاستعلام.فنسخة جدول البيانات "bookTable"هو السجل الأول الذي يبدو مثل نسخة IEnumerableمن خلال الطريقة AsEnumerable، ومن ثم،بما أن كل حقل مضمن في الاستعلام ،فإن نوع بياناته تم التصريح عنها من خلال شروط Ofالشمولية،متبوعة باسم الحقل ضمن علامات اقتباس.أخيراً،لأن،الاستعلام ليس لديه إمكانية وصول مباشرة إلى أسماء الحقول،فيتم إنشاء مجموعة النتائج باستخدام قاعدة "بادئ أو ممهد الكائن initializer ".إنها طريقة ملتوية roundaboutأكثر من لينكو على كائنات LINQ to Objects. ولكن إذا كان لديك بيانات تقبع في كائنات آدو دوت نت التي في الذاكرة،فإن لينكو على مجموعة بيانات هي الطريقة المناسبة.
يتضمن لينكو على مجموعة بيانات أيضاً دعم من أجل مجموعات البيانات "المحددة النوعtyped "،ومجموعات البيانات التي تتضمن توصيف البيانات metadataالضروري لوصف نوع البيانات كاملاً بالنسبة لكل حقل.مع مجموعات البيانات المحددة النوع،لاتحتاج بشكل مستمر إلى حفظ مساعدة لينكو من خلال الشروط "من نوع البياناتOf datatype"،سيعمل لينكو على استكشاف أنواع الحقول بشيء من الخصوصية.من أجل معلومات حول إنشاء مجموعات بيانات محددة النوع،راجع مستندات ميكروسوفت MSDNالتي تأتي مع الفيجوال أستوديو.

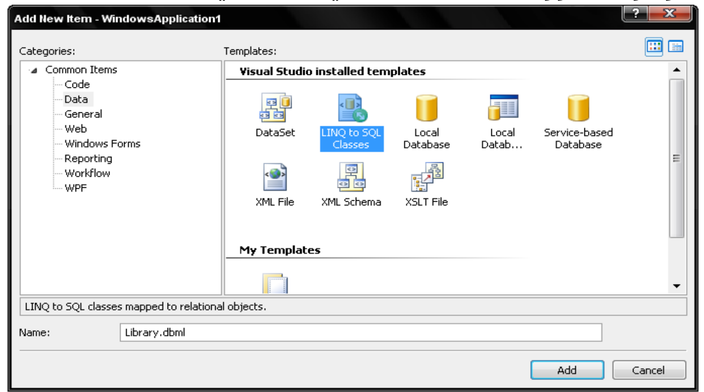
لينكو على سكول.LINQ to SQL

إن لينكو على سكول هو الموفر الذي يتيح لاستعلامات لينكو التفاعل مع قواعد بيانات سكول سرفر.بما أن مشروع المكتبة يستخدم سكول سرفر،سنمضي وقت أطول على هذه التقنية.كما مع لينكو على كينونات،فإن لينكو على سكول يعمل من خلال الفئات الوسيطة.على الرغم من أنك ومن المحتمل أن توفر عرض منطقي مختلف لجداول البيانات الجسدية باستخدام لينكو على سكول،يوجد أكثر من توقع لأن تكون كائنات لينكو على سكول أكثر شبهاً بجداول قاعدة البيانات المضمنة.

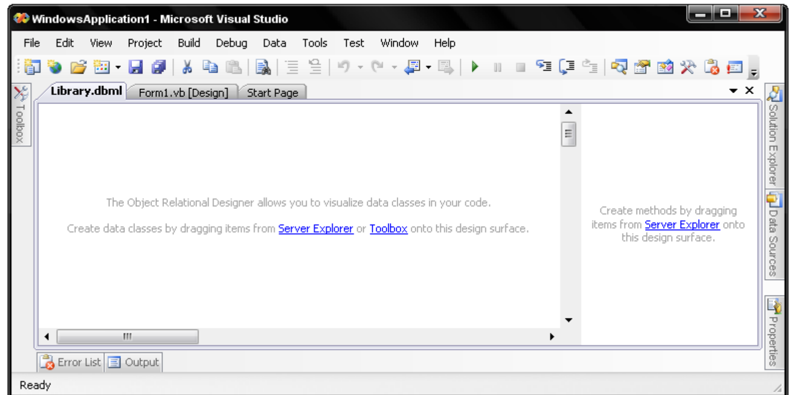
يتضمن لينكو على سكول أداة،مصمم الكائن العلائقي *Object Relational (O/R) Designer*، والذي سيساعدنا في إنشاء الفئات الوسيطة.تستطيع أخذ نظرة خاطفة على الشكل التالي لرؤية ما يبدو عليه.ما يزال يقوم بعمله المتوقع عند عمل اتصال قاعدة البيانات الضروري.المصمم العلائقي للكائنات هو مجرد سحب وإسقاط،وهو مناسب لقواعد البيانات التي لا تكون كبيرة بشكل فظيع.إذا احتجت إلى إنشاء فئات اتصال لقاعدة البيانات،والتي لديها،مثلاً، المئات من الجداول،فعليك اكتساب المعلومات عن الأداة *SqlMetal.exe* التي تأتي مع الفيجوال أستوديو.ستجد التفاصيل الكاملة في مستندات ميكروسوفت MSDN التي تأتي مع الفيجوال أستوديو.

استخدام لينكو على سكول يتم عمله بخمس خطوات سهلة.بإمكانك التتبع على طول في مشروع نماذج ويندوز جديد إذا أردت:

1.أضف ملف *dbml* جديد. هذا الملف عملياً هو القليل من الملفات حيث تعرضه الفيجوال أستوديو كملف واحد ـ يصف سياق بياناتك *data context*، والفئة العامة(الرئيسية master class)التي تحتوي كود الوصلة لكل جدول قاعدة بيانات الذي ستستخدمه في تطبيقك.لإنشاء هذا الملف من مشروع الفيجوال أستوديو،استخدم القائمة مشروع Project >> إضافة بند جديد Add New Item لعرض نموذج إضافة بند جديد.من "تصنيف البيانات Data category"، اختر قالب الفئات لينكو على سكولLINQ to SQL، غيّر اسم الملف من الافتراضي إلى Library.dbml، وانقر على الزر "إضافة Add ". كما في الشكل التالي.

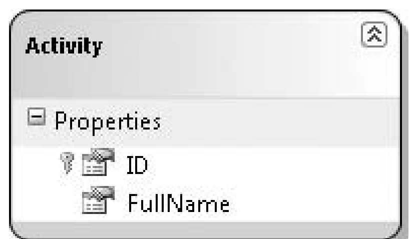


يظهر البند Library.dbml في مشروعك،والذي يفتح المصمم العلائقي للكائنO/R Designer، المبين في الشكل التالي.إذا تفحصت خاصياته،سترى أن اسمه LibraryDataContext.



2.أضف جداول إلى مصمم الكائن العلائقي.افتح سرفر إكسبلورر في الفيجوال أستوديو(اختر القائمة عرض View >> سرفر إكسبلورر Server Explorer)سترى بشكل مسبق وصلة إلى قاعدة بيانات المكتبة في قسم اتصالات البيانات Data Connections لشجرة سرفر إكسبلورر،بما أنني عملت على إنشاءها في فصول سابقة.فإنها ستدعى شيء مشابه لـ *myserver\sqlexpress.Library.dbo*.مدد ذلك التفرع من الشجرة،ومن ثم تفرع الجداول في أسفله.جميع الجداول في قاعدة بيانات المكتبة ستظهر.

اسحب وأسقط جدول "النشاط Activity " من سرفر إكسبلورر إلى النصف اليساري لمصمم الكائن العلائقي.عاجلاً أم آجلاً،ستظهر صورة الجدول على الشاشة شاهد الشكل التالي.



3.ابني تطبيقك،لقد وجدت أن هذه الخطوة ضرورية في بعض تركيبات installsالفيجوال أستوديو،ولكن ليس في بعضها الآخر.فهي تعيد تنشيط عرض الفيجوال بيسك لفئات LibraryDataContext الجديدة.لبناء التطبيق،اختر قائمة بناء Build >> بناء Build WindowsApplication1.

4.افتح سياق بياناتك الخاصة.الكود الناتج بواسطة مصمم الكائن العلائقي يعمل على تعريف التفاعل بين برنامجك وقاعدة البيانات،ولكن يبقى عليك تعين اتصال قاعدة البيانات عندما تشغل تطبيقك،أضف أداة زر إلى الفورمForm1، ومن ثم أضف الكود التالي إلى معالج حدث نقر الزر.

```
Dim LibraryDB As New SqlClient.SqlConnection("Data Source=myserver\sqlexpress;" & "Initial Catalog=Library;Integrated Security=true")
        Dim libraryLink = New LibraryDataContext(LibraryDB)
```

اعمل على تبديل *myserver* فيالكود باسم نظامك الخاص،وحدِث إعدادات الأمن إذا كنت تستخدم تصديق سكول سرفر SQL Server authentication .

5.كتابة الاستعلامات*write queries*،أنت الآن جاهز لتصميم استعلامات لينكو.إليك الكود الذي يحصل على النشاطات الخمسة الأولى من جدول النشاط Activity ويفرزها.

```
Dim activities = From act In libraryLink.Activities Where act.ID <= 5 Order By act.FullName
        For Each oneItem In activities
            MsgBox(oneItem.ID & ": " & oneItem.FullName)
        Next oneItem
        ' Messages --> 2: Manage author and name types
        ' 1: Manage authors and names
        ' 3: Manage copy status codes
        ' 4: Manage media types
        ' 5: Manage series
```

إذا نقرت الزر "أظهر جميع الملفات"في مستكشف الحلول،تستطيع الوصول إلى الملف dbml. الواقع تحت ملف المصمم،*Library.designer.vb* يحتوي هذا الملف على الفئات الوسيطة go-between classes المولدة المستخدمة من قبل لينكو على سكول.طالما أننا نستخدم جدول Activity في استعلامات لينكو،إليك الأجزاء ذات الصلة بالكود المصدري المولدة بشكل آلي:

```
<System.Data.Linq.Mapping.DatabaseAttribute(Name:="Library")> _
Partial Public Class LibraryDataContext
    Inherits System.Data.Linq.DataContext
    Public ReadOnly Property Activities() As System.Data.Linq.Table(Of Activity)
        Get
            Return Me.GetTable(Of Activity)()
        End Get
    End Property
End Class
<Table(Name:="dbo.Activity")> _
Partial Public Class Activity
    Private _ID As Long
    Private _FullName As String
    <Column(Storage:="_ID", DbType:="BigInt NOT NULL", _
    IsPrimaryKey:=True)> _
    Public Property ID() As Long
        Get
            Return Me._ID
        End Get
    End Property
    <Column(Storage:="_FullName", _
    DbType:="VarChar(50) NOT NULL", CanBeNull:=False)> _
    Public Property FullName() As String
        Get
            Return Me._FullName
        End Get
    End Property
End Class
```

تنفذ الفئة LibraryDataContext فئة سياق بيانات لينكو الخاصة LINQ data context class التي تبدو مشابهة للإصدار الأصغر من قاعدة بياناتي.فهي تحتوي على مراجع لهذه الجداول التي اخترت تضمينها في الربط،جميع جداول المكتبة ستظهر في هذه الفئة إذا كنت قد اخترتهم.لذلك،عندما أشرت إلى libraryLink.Activities في عينة استعلام لينكو LINQ،فإنه كان يشير إلى عضو النشاطات Activities العام من سياق البيانات.

يعرض جدول النشاط خاصيات مميزة تتطابق مع حقول قاعدة البيانات المضمنة.لذلك،لا مفاجأة إذا ما كنت قادر على الاستعلام عن هذه الفئات من خلال لينكو تماماً مثل ما فعلت مع فئة النوع لينكو على كائنات.ولكن يوجد جزء غريب حول كيفية حصول الفئة على البيانات من قاعدة البيانات.وهو الجزء المخفي لينكو على سكول LINQ to SQL، والمعالَج من خلال فئة DataContext القاعدية والمواصفات attributes المرافقة من فضاء الأسماء System.Data.Linq.Mapping .

ملاحظة:
ما وراء الكواليس،يعمل لينكو على سكول LINQ to SQL بشكل نظامي على إنتاج عبارات سكول للاستعلام وحتى لتحديث السجلات في جداول قاعدة البيانات الحقيقية.تستطيع تفحص هذه الاستعلامات الناتجة باستخدام أداة مظهر تصحيح استعلام سكول SQL Query Debug. إنها لاتأتي مع الفيجوال أستوديو،ولكن تستطيع تحميلها من موقع MSDN لميكروسوفت .

التنفيذ المؤجل.Deferred Execution

عندما تبني استعلام لينكو،لا تعالج الفيجوال بيسك الاستعلام مباشرةً،بالمقابل،تعمل على تأجيل التنفيذ *defers execution*، مشغلة الاستعلام عندما تطلب سجل من النتائج فقط .ويتيح لك هذا بناء استعلام من الأجزاء،وليس عليه استهلاك دورات المعالج CPU حتى تحتاج بشكل عملي البيانات النهائية.

```
' WARNING: Simplistic example.
        Dim someBooks = From bk In Library Select bk.Title, bk.Pages
        Dim orderedTitles = From bk In someBooks Order By bk.Title
```

في هذا الكود،ترتيب السجلات لايحدث حتى العبارة الثانية.ولكن هذا لا يعني شيء بما أنه لايوجد شيء تم معالجته بشكل عملي بواسطة العبارة الأولى.تذكر أن لينكو بشكل حقيقي يعمل فقط على تحويل استعلاماتك إلى طرق التوسيع وتعابير لامدا.الإسناد إلى someBooks يعمل شيء مشابه للتالي:

```
someBooks = Library.Select("Title, Pages")
```

أما الإسناد إلى orderedTitles فببساطة يوسع someBooks.

```
orderedTitles = Library.Select("Title, Pages").OrderBy("Title")
```

عملياً تحدث المعالجة عندما تطلب سجل من orderedTitles.بـواسطة "المعالجة processing"،أعني أن كل طريقة توسيع يتم تنفيذها على مصدر بيانات المكتبة الأصلي بالترتيب،من اليسار إلى اليمين.من أجل orderedTitles،يتم تخفيض بيانات المكتبة الأصلية من خلال الطريقة "Select "،ومن ثم يتم تعديلها إلى حدٍ أبعد بواسطة الطريقة OrderBy.

تضمين طرق يتم معالجتها من اليسار إلى اليمين تشرح لما ترتيب شروط of مثل Skip وTake هامة جداً.فالتعبير :

```
Library.Take(2).Skip(2)
```

مختلف عن التالي:

```
Library.Skip(2).Take(2)
```

مشروع. Project

في النهاية سنعمل في هذا الفصل على إضافة ما يعتبر قلب نظام المكتبة:البحث lookupعن الكتب وبنود المكتبة الأخرى من قبل الزبائن.

بحث بنود المكتبة.Looking Up Library Items

عندما عملنا على بناء نموذج المكتبة الرئيسي في الفصل 7،عملنا على تضمين حقول تتيح للزبون البحث عن بنود المكتبة.ولكن كان ذلك كل ما عملناه،فلم نعمل على تمكين الحقول أو جعلها قابلة للاستخدام.ولم نضمن أيضاً أي مكان لعرض قائمة بالبنود المتطابقة.لنعمل على إتمام هذه المكونات في هذا الفصل.سنبدأ مع قائمة البنود المتطابقة.

لقد عملت على إضافة فورم إلى المشروع اسمها*ItemLookup.vb* والتي تعرض نتائج بحث بنود المكتبة.وتتضمن عدة أزرار عند الحافة العلوية للفورم،وثلاث لوحات عرض رئيسية:

PanelMatches

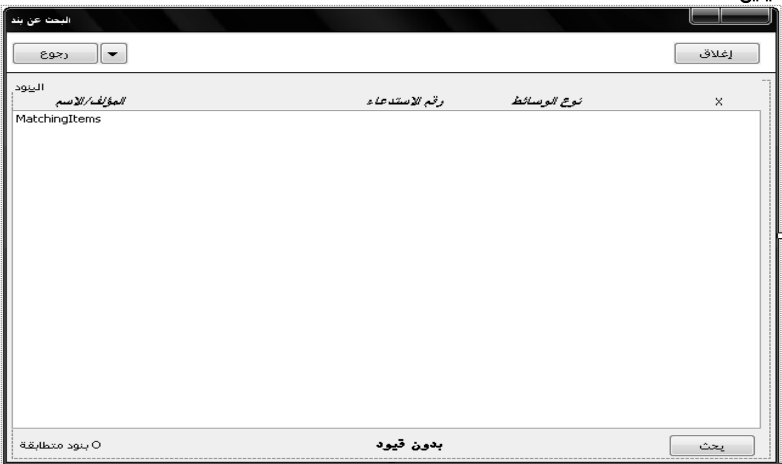
تحتوي صندوق قائمة listboxضخم يعرض التطابقات لغير البنود.على سبيل المثال،إنها تعرض قائمة من أسماء المؤلفين والناشرين المتطابقة كما تم البحث عنها من قبل زبون.
عندما تظهر هذه اللوحة،يختار الزبون نموذج مطابقة القائمة MatchingGeneral،وينقر على الزر "بحث" لعرض البنود المرتبطة إلى المؤلف ،الناشر أو مدخلة أخرى.

PanelItem

تحتوي صندوق قائمة ضخم يعرض البنود من جدول قاعدة البيانات NamedItem.فهي تعرض قائمة ببنود المكتبة التي تطابق معيار ما.اختيار بند ما من قائمة MatchingItems والنقر على الزر "بحث" يعرض تفاصيل ذلك البند .

PanelOneItem

تحتوي على أداة عرض ويب WebBrowserوالتي تعرض التفاصيل حول بند مكتبة وحيد .محتوى التفاصيل يتم بناءه باستخدام HTML قياسي،ويمكن أن يحتوي وصلات تعيدك إلى لوحة PanelItemsمع مجموعة جديدة من البنود المتطابقة المعروضة.على سبيل المثال،إذا كنت تعرض تفاصيل كتب الفيجوال بيسك2008 الفائزة بالجوائز ونقرت على اسم الناشر لذلك البند،تظهر اللوحة PanelItems،مجدولة جميع البنود المعمولة بواسطة ذلك الناشر.
تتضمن الفورم أيضاً مجموعة من الأزرار الرجوع(في الزاوية العلوية اليمينية)والتي تعمل مثل أزرار الرجوع في مستعرض الويب خاصتك،زر الإغلاق يغلق هذه الفورم ويرجع التركيز إلى الفورم الرئيسي،وقائمة(قائمة الرجوعBackMenu)تستخدم لدعم ميزة زر العودةBack .يبين الشكل التالي الفورم مع لوحة PanelItemsالظاهرة في المقدمة،بما أنها تظهر أهمية أكثر بقليل من اللوحتين الباقيتين.



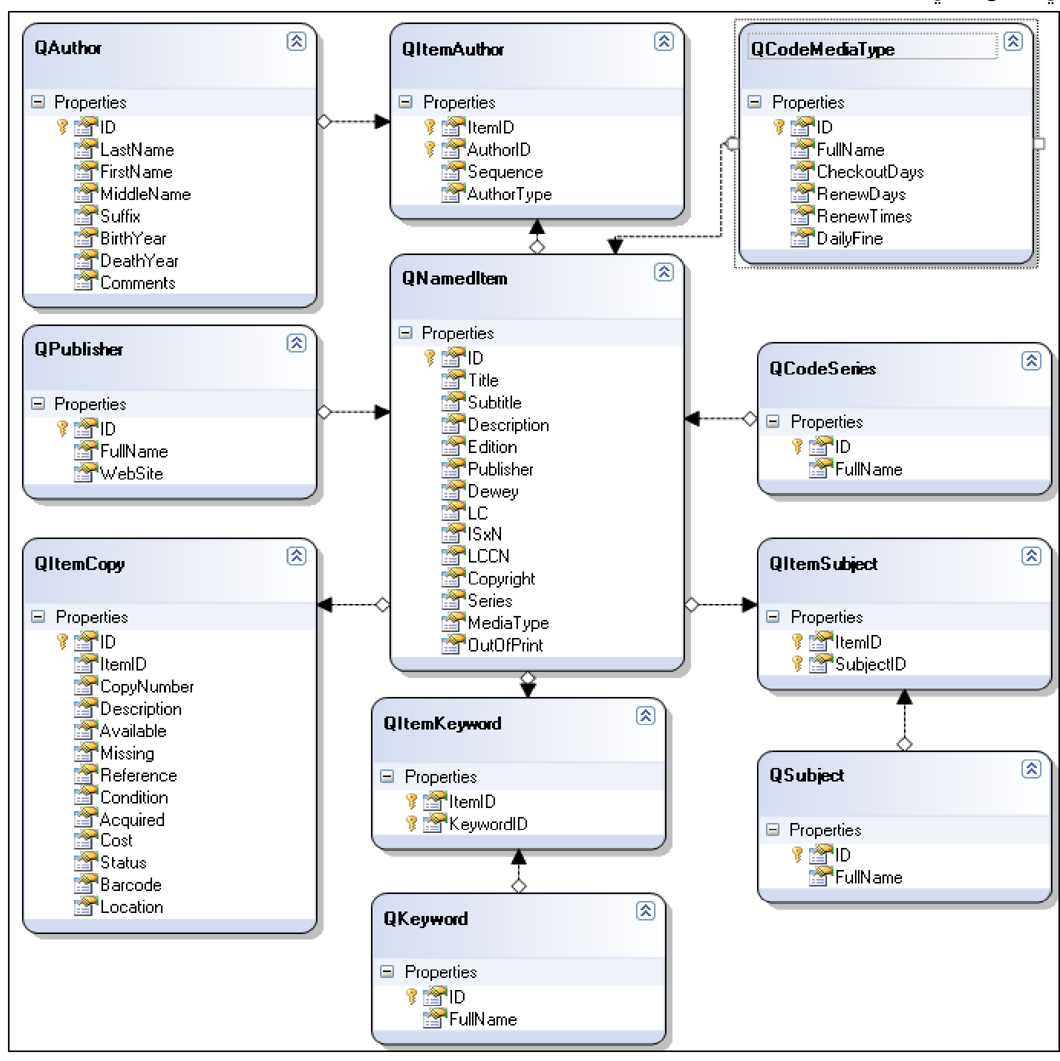
معظم كود هذه الفورم يركز على تعبئة كل من محتوى صناديق القائمة listboxesوتفاصيل HTML.هذا البحث يتم عمله على الفورم الرئيسية والذي يستدعي ضمن نموذج البحث هذا الطريقة InitiateSearch.بحث قاعدة البيانات الفعلي عن البنود المتطابقة يحدث في الطريقة PerformLookup،والتي يتم استدعاءها من قبل InitiateSearch.تتضمن الطريقة PerformLookup استعلامات لينكو التي ترحل إلى قاعدة بيانات المكتبة وتعود بواسطة الموفر لينكو على سكول.تتضمن الاستعلامات من أجل أنواع البحث المختلفة التي تم تضمينها:أبحاث حسب العنوانtitle،المؤلفauthor،الموضوعsubject،الكلمة المفتاحيةkeyword،الناشرpublisher،السلاسلseries، كود التعريف bar code،وبعض أرقام المعرفات ID number.معظمها للاستخدام الداخلي.نوع البحث الذي تم عمله يحدد أي من اللوحات الثلاث سيتم عرضها(بواسطة المتغير resultType).
فبحث المؤلف يعرض اللوحة PanelMatchesمع قائمة بأسماء المؤلفين الموافقة،يعرض بحث العنوان البنود المتطابقة على لوحة PanelItems.قبل أن نلقي نظرة على كود لينكو،نحتاج إلى تثبيت مجموعة من الأشياء في باقي التطبيق لدعم استعلامات لينكو الثلاث هذه.لذلك عملت على عدم تمكين الملف *ItemLookup.vb* من الترجمة الآن لأن عمله يؤدي إلى توليد أخطاء.
بقدر كون لينكو على سكول مذهل ،فإنه يبقى بحاجة إلى لمسة الإنسان(وهو أنت) للمساعدة على إيجاد جداول قاعدة بيانات سكول سرفر.سنعمل على استخدام مصمم الكائن العلائقي Object Relational Designer الذي عملنا معه سابقاً في هذا الفصل.اختر من القائمة مشروع Project >> إضافة بند جديدAdd New Item . على نموذج إضافة بند جديد،اختر بيانات Dataمن قائمة التصنيفاتCategories،اختر الفئات لينكو على سكول LINQ to SQL Classes من حقل القوالب Templates. وضع الاسم في حقل الاسم إلى Library.dbml قبل النقر على الزر إضافة Add.وبالنقر على زر إضافة تظهر نافذة المصمم الفارغةO/R Designer .
افتح السرفر إكسبلورر Server Explorer واستعرض قاعدة بيانات المكتبة*Library* .ومن تفرع الجداول Tables،اسحب واسقط الجداول التالية على النصف اليساري من نافذة المصمم الكائن العلائقي:

- Author
- CodeMediaType
- CodeSeries
- ItemAuthor
- ItemCopy
- ItemKeyword
- ItemSubject

- Keyword
- NamedItem
- Publisher
- Subject

سيحلل المصمم بشكل صحيح العلاقات بين الجداول ويظهر خطوط الربط بين المراجع الثانوية.تستطيع ترتيب الجداول عند الحاجة لرؤية الجداول بشكل أفضل، لنعمل على إعادة تسمية الجداول شيء ما،يحاول المصمم أن يكون زكي حقاً،مغيراً أي جمع لأسم جدول موجود إلى مكافئه المفرد .(بشكل تقليدي الأسماء المفردة مفضلة عند تصميم جداول قاعدة البيانات)،لسوء الحظ،لقد خرب التحويل اسم جدول CodeSeries،مغيراً إياه إلى CodeSery.إنه جيد،ولكن ليس له معنى.اختر ذلك الجدول وغير خاصية اسمه Nameإلى CodeSeriesفي نافذة الخصائصProperties .

يعمل ذلك على إعادة وضع أسماء الجداول إلى جذورها،ولكنه ما يزال غير جيد،المشكلة أننا استخدمنا بعض أسماء هذه الجداول لأسماء النماذج أو الفورمات في تطبيق المكتبة. الفئات المتضاربة في فضاءات أسماء مختلفة ،لذلك يمكن أن تتم ترجمة الكود،ولكن يبقى علينا كتابة الكثير من فضاءات الأسماء عند تحديد فضاءات الأسماء هذه ،بحيث يمكن ترجمة الكود،وبما أنني كسول جداً لعمل ذلك.لإزالة التضاربات،قررت إضافة الحرف *Q* إلى كل اسم جدول لينكو على سكولLINQ to SQL. في المصمم،اختر كل جدول واعمل على تسميته من جديد، مضيفاً *Q* إلى بداية كل جدول بحيث مثلاً يصبح اسم الجدول CodeSeries بالاسم QCodeSeries وهكذا.عندما تنتهي فيجب أن يكون لديك عرض المصمم مشابه للمعروض في الشكل التالي.



على الرغم من صعوبة ضمان تجنب التعارضات لجميع الأسماء وأنها مفردة،فعندما نستخدم بيانات المكتبة في سياق استعلامات لينكو،سنجد أن جميع أسماء الفئات من أجل لينكو على سكول المولد لهذه الجداول قد جمع الأسماء (أي QPublishersبدلاً عن QPublisher).

بالرجوع إلى كود المشروع من أجل الفصل 12،فقد عملنا على إضافة طريقة توسيع إلى الفئة SqlClient.SqlDataReader والتي تنسق اسم المؤلف من استعلام قاعدة البيانات.

```
<System.Runtime.CompilerServices.Extension()> _
    Public Function FormatAuthorName(ByRef dbInfo As SqlClient.SqlDataReader) As String
```

لسوء الحظ هذا الروتين مفيد فقط مع كائناتSqlDataReader. في الروتين PerformLookupالذي سنعمل على إضافته،نحن بحاجة إلى تنسيق أسماء المؤلفين من استعلام لينكو لسجلات جدول QAuthor.أكاد أخمن أننا بحاجة لطريقة توسيع أخرى لنوع الكائن ذاك.افتح الكود المصدري للوحدة البرمجية *General.vb* وأضف الطريقة الجديدة FormatAuthorNameإليها .

```
<System.Runtime.CompilerServices.Extension()> _
   Public Function FormatAuthorName(ByVal author As QAuthor) As String
      ' تقديم سجل مؤلف,للفوددة بالاسم المنسق
        Dim authorName As String
        On Error Resume Next
        ' تنسيق الاسم
        authorName = CStr(author.LastName)
        If (author.FirstName IsNot Nothing) Then
            authorName &= ", " & author.FirstName
            If (author.MiddleName IsNot Nothing) Then authorName &= " " & author.MiddleName
        End If
```

```
        If (author.Suffix IsNot Nothing) Then _
            authorName &= ", " & author.Suffix
        ' أضف سنوات الميلاد والموت
        If (author.BirthYear IsNot Nothing) Or _
                (author.DeathYear IsNot Nothing) Then
            authorName &= " ("
            If (author.BirthYear Is Nothing) Then
                authorName &= "????"
            Else
                authorName &= CStr(Math.Abs(CInt(author.BirthYear)))
                If (author.BirthYear < 0) Then authorName &= "BC"
            End If
            authorName &= "-"
            If (author.DeathYear IsNot Nothing) Then
                authorName &= CStr(Math.Abs(CInt(author.DeathYear)))
                If (author.DeathYear < 0) Then authorName &= "BC"
            End If
            authorName &= ")"
        End If
        ' في النهاية
        Return authorName
    End Function
```

إذا قارنت هذا الكود المصدري مع نسخة SqlDataReader،ستجد أن هذه النسخة أكثر وضوحاً بما أنها تشير إلى أعضاء الفئة بدل حقول قاعدة البيانات من خلال القارئ.وهذا بفضل لينكو.

من أجل تغيرات دعم لينكو،مكّن الملف *ItemLookup.vb* باختياره من مستكشف الحلول وغير خاصية Build Action من None إلى Compile.والآن لنرجع إلى الكود في ذلك الملف.

يؤلف الروتين PerformLookup معظم عبارة الشرط If الضخمة،مع شروط مختلفة من أجل معظم أنواع البحث المختلفة.يعالج شرط Else الأخير جميع الأبحاث التي ستملأ القائمة على لوحة الفورم PanelItems .تلك القائمة التي تظهر البنود بشكل عملي.فهي تحوي الكثير من عبارات الشرط If ولكن ما هو هام هو استعلام لينكو.بدل من أن يكون فقط مجرد استعلام بسيط،فهو استعلام معقد ينمو قليلاً قليلاً.يبدأ الاستعلام بالقواعد،وطلب الكثير من السجلات من جدول قاعدة البيانات NamedItem.(المتغير libraryDC هو سياق البيانات المفتوحة من أجل قاعدة بيانات المكتبة *Library*).

```
Dim itemQuery = From ni In libraryDC.QNamedItems
```

التالي إذا طلب المستخدم بنود نوع وسيطة معينة("أظهر لي فقط الديفيدي DVDs المتطابقة،وليس الكتب")،يتم تحديث الاستعلام بمحاكاة الشرط Where.

```
If (LimitByMedia <> -1) Then
' التحديد لنوع وسيطة معينة
            itemQuery = From ni In itemQuery Where ni.MediaType = LimitByMedia
        End If
```

يضبط نوع البحث أيضاً الاستعلام.على سبيل المثال،بحث الكلمة المفتاحية keyword يضيف كلمات محددة من قبل المستخدم كمعيار:

```
'عمل بحث لأي كلمة مفتاحية
 keywordSet = New Generic.List(Of String)
 keywordSet.AddRange(Split(searchText.ToUpper, ","))
 itemQuery = From ni In itemQuery _
    Let keySet = (Aggregate ik In ni.QItemKeywords _
         Into Any(keywordSet.Contains(ik.QKeyword.FullName.ToUpper))) _
            Where keySet = True Select ni
```

تلك الإضافة تستخدم الاستعلام الجزئي للإجمالي ضمن الاستعلام الرئيسي.الكلمة المحجوزة Let والتي هي جزء من لينكو،تعمل على إسناد استعلام جزئي أو نوع نتيجة آخر إلى متغير مؤقت ضمن الاستعلام(keySet في هذه الحالة)لذلك يمكن أن يتم الإشارة إليه في مكان آخر في الاستعلام.

حالما يتم إضافة شروط Where،يتم تخزين واستخدام كامل الاستعلام.

```
'جميع استعلامات بند يتم فرزها بواسطة العنوان
        itemQuery = From ni In itemQuery Order By ni.Title, ni.Subtitle
```

بعض استعلامات لينكو في الروتين بسيطة جداً،إليك الكود الذي يعمل بحث عن اسم الناشر:

```
'تحضير الاستعلام من أجل بحث الناشر
holdText = Trim(searchText)
 If (InStr(holdText, "*") = 0) Then holdText &= "*"
Dim publisherQuery = From pb In libraryDC.QPublishers Where pb.FullName Like holdText Order By
pb.FullName
```

لا يبدو أنه مختلف عن ما نتوقعه تماماً في استعلام سكول.وشيء أخر مهم وهو أن القيم الشاملة wildcards تستخدم الحرف* بدل الرمز% في استعلام سكول القياسي.بعد معالجة هذا الاستعلام، يتم عمل مسح على نتائج لينكو،ونقل السجلات إلى القائمة MatchingGeneral.

```
For Each publishItem In publisherQuery
' التقييد إلى السجلات الأعظمية المحددة
                If (matches >= SearchMatchLimit) Then
                    matches += 1
                    Exit For
                End If
' بند قائمة عام
                MatchingGeneral.Items.Add(New ListItemData( _
                    publishItem.FullName, CInt(publishItem.ID)))
```

```
                matches += 1
            Next publishItem
```

هذا مشابه لكود رأيته سابقاً في فصول سابقة.فهو يحمل أداة صندوق قائمة ListBox بالكائنات ListItemData،كل منها يحتوي عرض اسم ورقم معرف من قاعدة البيانات.وهذه جميل من أجل متطلبات عرض بسيط.

ولكن إذا رجعت وألقيت نظرة على الشكل السابق(شكل "نموذج البحث عن بند" *ItemLookup*)،من الواضح أننا نريد شيء ما أكثر أهمية من أجل قائمة البنود المتطابقة.نريد أعمدة،والأعمدة تتطلب بيانات معقولة أو ممكنة.

لتخزين هذه البيانات سنعمل على تركيب فئة جديدة،تدعى MatchingItemData،والتي تعمل تماماً مثل ListItemData،ولكن لديها حقول بيانات أكثر.

```
' فئة لقنص بنود في قائمة تطابق بنود
    Private Class MatchingItemData
        Public ItemID As Integer  ' NamedItem.ID
        Public Title As String
        Public Subtitle As String
        Public Author As String
        Public MediaType As String
        Public CallNumber As String
        Public Overrides Function ToString() As String
            ' بناء نص العرض.
            If (Subtitle = "") Then
                Return Title & ", by " & Author
            Else
                Return Title & ": " & Subtitle & ", by " & Author
            End If
        End Function
    End Class
```

بما أن هذه الفئة سيتم استخدامها لعرض البنود المتطابقة على هذه الفورم فقط،فقد جعلتها فئة تابعة ضمن فئة الفورم ItemLookup الأكبر.تعمل الطريقة ToString على إخراج النص الذي يظهر في القائمة. لن نعمل على إنتاج مخرجات عمودية عملياً حتى الفصل اللاحق.أما الآن،سنعمل على عرض العنوان والمؤلف فقط.

كل من اللوحة PanelMatches واللوحة PanelItems يتضمنان الزر بحث والذي يمهد لاستدعاء جديد لـ PerformLookup بالاعتماد على البند المختار في القائمة.زر البحث على اللوحة يستخلص الكائن المختار من القائمة MatchingItemData، ويعمل بحث جديد.

```
private Sub ActItemLookup_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ActItemLookup.Click
' البحث عن بند بواسطة المعرف المحدد
        Dim itemID As Integer
' تجاهل إذا لم يتم اختيار تطابق
        If (MatchingItems.SelectedIndex = -1) Then Return
        itemID = CType(MatchingItems.SelectedItem, MatchingItemData).ItemID
' عمل بحث
        If (PerformLookup(LookupMethods.ByDatabaseID, CStr(itemID), False) = False) _
            Then Return
' تخزين التاريخ
        AddLookupHistory(LookupMethods.ByDatabaseID, CStr(itemID))
    End Sub
```

استدعاء الروتين PerformLookup يبدأ بمعالجة شاملة مرةً أخرى.

المحافظة على تاريخ البحث Maintaining Search History

إذا كان لدينا زبون لديه وقت فراغ كبير،ويريد البحث عن كتاب "الحرب والسلم *war and Peace*"

. يبدأ من "البحث التمهيدي InitiateSearch" وينتقل إلى كود "عمل بحث PerformLookup "،العنوان الأولي"الحرب والسلم"يعرض قائمة من العناوين المتطابقة على اللوحة PanelItems.

. يحدد الزبون الكتاب في هذه القائمة،وينقر زر البحث،والذي يستدعي معالج الحدث ActItemLookup_Click.

. معالج الحدث هذا يستدعي مرةً أخرى PerformLookup، وهذه المرة يعمل بحث دقيق بالاعتماد على معرف ID قاعدة البيانات ضمن الجدول NamedItem .

. تظهر تفاصيل بند على اللوحة PanelOneItem (وسأناقش كيفية عملها فيما بعد في هذا الفصل).

. تتضمن التفاصيل وصلة إلى الكاتب الروسي "تلستوي،ليو"مؤلف الكتاب ذو المعاناة الطويلة.عندما ينقر الزبون على هذه الوصلة،فإنها تمهد لاستدعاء أخر للروتين PerformLookup،وهذه المرة بواسطة معرف ID المؤلف.

. نعود إلى اللوحة PanelItems،نعرض قائمة كتب وبنود أخرى للكاتب الروسي تلستوي Tolstoy، على فرض أن لديه الوقت ليكتب أي شيء أخر.

إذاً الزبون لديه الآن خبرة بلوحات البحث هذه:(1) قائمة "عامة"بالعناوين المتطابقة لأسم "الحرب والسلم".،(2)يتم عرض "التفاصيل"للبند "الحرب والسلم" المختار،و(3) قائمة "بنود"كتاب المكتوبة بواسطة ليو تلستوي Leo Tolstoy. ميزة التاريخ(أو البنود الأخيرة المفتوحة)المضمنة في هذه الفورم تتيح للزبون العودة إلى أي صفحة بحث سابقة،تماماً مثل الميزة الموجودة في متصفح الانترنت الذي لديك.

من الممكن أن ينتج عن بعض الأبحاث المئات المعمولة من النتائج .فنحن لانريد تخزين جميع المحتوى في الذاكرة،بما أنه من المحتمل أن لاينقر المستخدم أبداً على الزر "رجوع". بالمقابل،سنعمل على عمل ما يعمله متصفح الانترنت تماماً:تخزين المعلومات الأقل المطلوبة لعمل استعلام مرةً أخرى.فمتصفح الانترنت يحتفظ فقط بالاسم وعنوان الانترنت URL للمسارات التي تم زيارتها في قائمة "العودة".(تخزين الملفات والصور ليست جزء من ميزة التاريخ)،تحتاج الفورم *ItemLookup.vb* تخزين فقط هذه القيم التي يحتاجها الروتين PerformLookup لعمل بحث مرةً أخرى:نوع البحث،ومعيار البحث النصي أو العددي المستخدم في البحث.

يتم الوصول إلى تاريخ الزبون بالقاعدة "ما يدخل أخيراً يخرج أولاً last-in, first-out "فالصفحة الأحدث التي تم عرضها،هي التي يريد الزبون رؤيتها أولاً عند استخدام الزر "رجوع"،ناقشنا التركيب"مايدخل أخيراً يخرج أولاً، في الفصل 16:في الستاك(الذاكرة العشوائية).كل مرة يعرض المستخدم لوحة،سنعمل حاشية عليها،عارضين فقط تلك القيم التي سنحتاجها فيما بعد على الستاك(الذاكرة العشوائية).عندما يريد المستخدم عرض التاريخ،سنعرض محتوى البحث الأكثر حداثةً للذاكرة العشوائية ونحدث العرض.

تخزن الفئة ItemLookupHistory،وهي فئة أخرى تابعة ضمن الفئة ItemLookup،القيم التي نحتاجها لإدارة التاريخ في الستاك stack .

```
' فئة لقنص تاريخ العروض في نسخة الفورم هذه
    Private Class ItemLookupHistory
```

```
        Public HistoryDisplay As String
        Public LookupType As Library.LookupMethods
        Public LookupData As String
    End Class
```

يوفر الحقل HistoryDisplay اسم عرض قصير لمساعدة المستخدم على عمل بحث خلال التاريخ.بينما LookupType وLookupData هي قيم يتم تمريرها إلى PerformLookup. لجعل الأشياء أفضل،سنستخدم ستاك(أو ذاكرة عشوائية) شاملة generic stack من أجل التخزين الفعلي.وتم التصريح عنها كحقل للفئة ItemLookup.

```
Private LookupHistorySet As Collections.Generic.Stack(Of ItemLookupHistory)
```

كلما زار الزبون كل لوحة،تملأ الاستدعاءات إلى الطريقة الستاك بكل بند تم زيارته حديثاً.

```
Private Sub AddLookupHistory(ByVal searchType As Library.LookupMethods, ByVal searchText As String)
        ' إضافة بند إلى تاريخ البحث
        Dim newHistory As ItemLookupHistory
        Dim displayText As String
        ' بناء نص لعرض في بند جديد
        displayText = BuildDisplayText(searchType, searchText)
        ' بناء بند تاريخ جديد
        newHistory = New ItemLookupHistory
        newHistory.LookupType = searchType
        newHistory.LookupData = searchText
        newHistory.HistoryDisplay = displayText
        LookupHistorySet.Push(newHistory)
        ' تحديث زر العودة
        RefreshBackButtons()
    End Sub
```

فيما بعد عندما ينقر المستخدم على واحد من أزرار العودة،يختبر معالج حدث النقر BackMenuItems_Click تاريخ الستاك،ويستدعي PerformLookup عند الحاجة.ولأننا عملنا على تخزين الكائنات في ستاك شاملة generic stack، فليس علينا تخصيص تحويلها من System.Object،فالبرنامج يعرف تماماً ما هو نوع البيانات التي عليه.

```
Private Sub BackMenuItems_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
            Handles BackMenu1.Click, BackMenu2.Click, BackMenu3.Click, BackMenu4.Click, _
            BackMenu5.Click, BackMenu6.Click, BackMenu7.Click, BackMenu8.Click, _
            BackMenu9.Click, BackMenu10.Click
        ' واحد من بنود قائمة الرجوع تم نقره
        Dim whichItem As Integer
        Dim counter As Integer
        Dim scanHistory As ItemLookupHistory
        ' تحديد البند الذي تم نقره
        whichItem = CInt(DigitsOnly(CType(sender,  _
            System.Windows.Forms.ToolStripMenuItem).Name))
        If (whichItem >= LookupHistorySet.Count) Then Return
        ' التخلص من البنود الوسيطة
        For counter = 1 To whichItem
            LookupHistorySet.Pop()
        Next counter
        ' عمل بحث عند الطلب
        scanHistory = LookupHistorySet.Peek
        If (PerformLookup(scanHistory.LookupType, scanHistory.LookupData, _
            False) = False) Then Return
        RefreshBackButtons()
    End Sub
```

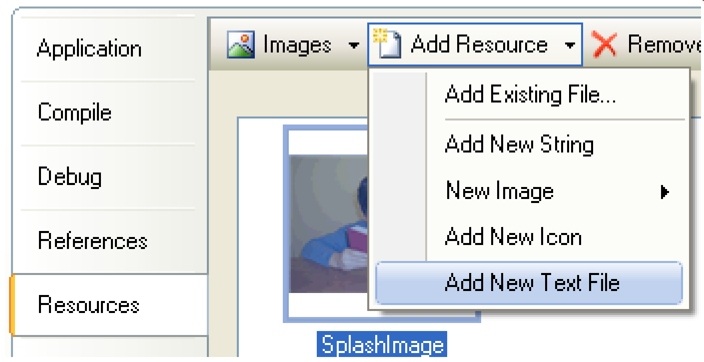
إظهار تفاصيل بند.Showing Item Detail

تبني الدالة BuildHTMLAndLinks محتوى HTML الذي يظهر على اللوحة PanelOneItem. تتضمن هذه اللوحة تفاصيل بند واحد SingleItemDetail، أداة مستعرض انترنت WebBrowser مضمنة مع الدوت نت.بشكل أساسي هي نسخة من مستعرض الانترنت Internet Explorer الذي تضمنه في تطبيقاتك.من الطبيعي،أن تزود لها عناوين الانترنت URL من أجل العرض،ولكن تستطيع توفير محتوى مخصص من خلال الخاصية DocumentText لأداة مستعرض الانترنت. طرق البحث ByDatabaseID وByBarcode ضمن الروتين PerformLookup تعمل على إسناد هذه الخاصية ضمن المحتوى المعاد من BuildHTMLAndLinks.

```
SingleItemDetail.DocumentText = BuildHTMLAndLinks(CInt(idQuery.ToArray(0)))
```

المحتوى الموفر بواسطة هذا الروتين هو HTML قياسي،ولكن مع بعض الوصلات المصنوعة بشكل خاص والتي تتيح لبرنامج المكتبة عمل بحث إضافي بالاعتماد على تفاصيل بند المكتبة المعروض.

معظم نص HTML معياري،ويبدو من المخزي إشغال نفسك في عمل ضم سلاسل حرفية لتضمين هذا المحتوى،لذلك وبالمقابل،عملت على تخزين الكثير من HTML كمصدر ملف نصي من خلال لوحة المصادر Resources لخاصيات المشروع properties. على تلك اللوحة،انقر على زر "إضافة مصدر Add Resource"،انقر على بند القائمة إضافة ملف نصي جديد Add New Text File، وأدخل ItemLookupBody كأسم لملف النص الجديد،كما هو مبين في الشكل التالي.



في نافذة محرر النص الذي يظهر،أضف محتوى HTML التالي:

```
<html>
<head>
<style type="text/css">
body { font-family: "Arial"; }
h1 { font-family: "Arial"; margin-top: 0px; margin-bottom: 0px; font-size: 18pt; font-weight: bold; }
h2 { font-family: "Arial"; margin-top: 20px; margin-bottom: 0px; font-size: 15pt; font-weight: normal; }
h3 { font-family: "Arial"; margin-top: 0px; margin-bottom: 0px; font-size: 15pt; font-weight: normal;
font-style: italic;  }
p { margin-top: 2px; margin-bottom: 2px; margin-left: 15px; font-family: "Arial"; font-size: 12pt; }
table { border: solid black 1px; margin-left: 15px; }
th { border: solid black 1px; background-color: black; color: white; white-space: nowrap; text-align:
left; }
td { border: solid black 1px; white-space: nowrap; }
a:visited { color: blue; }
</style>
</head>
<body>
```

إذا كنت تعرف HTML،ستميز معظم المحتوى كسياق قالب صفحة تخطيطي(مؤلف من تنسيق تخطيط وخط)Cascading Style Sheet (CSS) مغلف.قواعده التنسيقية المتنوعة ستجلب شكل ومضمون خاص ومستقر على محتوى المتصفح browserالذي يظهر ضمن نموذج البحث عن بند..وهذا ليس كتاب حول CSS،ولكن توجد بعض الكتب الجيدة تستطيع مراجعتها في خزائنك أو أن تبحث في الانترنت عن كتب بهذا الخصوص إذا كنت مهتم.

تستطيع إيجاد جزء محتوى HTMLفي مستكشف الحلول Solution Explorer، ضمن تفرع المصادر Resources، من المحتمل أنك لاحظت أنني لم أعمل على تضمين إغلاق الوسم </body> والوسم </html>،لقد عملت على إرفاق هذين الإغلاقين في الطريقة BuildHTMLAndLinks. بما أن ضم السلاسل الحرفية string concatenation مشهورة ببطئها،اخترت استخدام الفئة "باني النص StringBuilder" فئة شبيهة بالسلسلة النصية خاصة مصممة بشكل خاص من أجل السرعة عند إضافة محتوى بشكل متكرر على النص القاعدي.تعمل على إرفاق المحتوى إلى نهاية "باني النص StringBuilder" باستخدام الطريقة Appendوالطريقة AppendLine، وتستخلص النص الكامل من خلال الطريقة ToStringالقياسية .

سنبدأ المحتوى بنص HTMLالمعياري المجدول سابقاً.بما أننا أضفناه كمصدر resource، فإنه سيظهر في الكائن My.Resources تحت الاسم الذي منحناه إياه.

```
Dim detailBody As New System.Text.StringBuilder
detailBody.Append(My.Resources.ItemLookupBody)
```

معظم الكود يعمل على إضافة نص سهل إلى باني النص detailBodyباستخدام طريقته AppendLine،إليك الكود الذي يضيف عنوان الكتاب الرئيسي:

```
'إدخال العنوان
        sqlText = "SELECT Title, Subtitle FROM NamedItem WHERE ID = " & itemID
        dbInfo = CreateReader(sqlText)
        dbInfo.Read()
        detailBody.AppendLine("<h1>" & HTMLEncode(CStr(dbInfo!Title)) & "</h1>")
```

الدالة HTMLEncode المستدعاة في هذا المقطع،مضمنة في الفئة ItemLookup.فهي تعمل بعض التعديل البسيط لحروف معينة عند الحاجة بواسطة HTML.يتم استدعاءها بشكل متكرر من خلال BuildHTMLAndLinks.

لذلك ،هذا هو الـHTML ،ولكن ماذا بخصوص الوصلات؟إذا وضعت وصلة قياسية standard link،كالوصلة: *http://www.microsoft.com* فإن المستعرض المضمن سينتقل إلى الصفحة عند النقر على الوصلة.ولكن هذا لا يساعدني في عمل بحث على قاعدة البيانات.لا تعرض أداة WebBrowser "تم نقر الوصلة link clicked" بشكل حقيقي،ولكن لديها حدث التصفح Navigatingوالذي يكون مغلق.ينطلق هذا الحدث كلما أراد المتصفح الانتقال إلى صفحة جديدة.لحسن الحظ،واحدة من قيم البيانات الممررة لمعالج الحدث هو عنوان الانترنت URL المستهدف.لذلك،كل ما علينا عمله هو بناء وصلة تحتوي معلومات نحتاجها لعمل بحث قاعدة البيانات.

قررت تخزين تفاصيل بحث قاعدة البيانات المناسبة كتجمع collection (مشابه للذاكرة العشوائية للتاريخ history stack)،وأنشئت وصلات مشابهة لعناوين الانترنت URL مزيفة تشير إلى أي بند سيستخدم في التجمع. بعد الكثير من التفكير والتمعن،قررت استخدام تنسيق وصلات عناوين الانترنت URLالمزيفة:

library://x

حيث *x*يتم استبدالها بأي فهرس ضمن تجمع الوصلاتcollection. فهو بسيط ويعمل بشكل جيد،تجمع تفاصيل البحث هي قاموس شمولي generic dictionary مخزن كحقل ضمن فئة الفورم.

```
'فئة لقنص شبه الوصلات على عرض بند وحيد
    Private Class SingleItemLink
        Public LinkType As Library.LookupMethods
        Public LinkID As Integer
    End Class
Private ItemLinkSet As Collections.Generic.Dictionary(Of Integer, SingleItemLink)
```

ومن ثم بالعودة إلى كود بناء الـ HTML،عملت على إضافة عناوين انترنت URLsمزيفة وكائنات SingleItemLinkبشكل مترادف.إليك بعض الكود المستخدم لإضافة وصلات المؤلف،تقديم قارئ بيانات مع حقول اسم المؤلف.(توفر القيمة entryID بدل *x*في المكتبة.library://*x*)

```
Do While dbInfo.Read
' إدخال اسم هذا المؤلف
                holdText = dbInfo.FormatAuthorName()
                entryID += 1
                detailBody.AppendLine("<p><a href=""المكتبة://" & entryID & """>" & _
                    HTMLEncode(holdText & " [" & CStr(dbInfo!FullName) & "]") & _
                    "</a></p>")
' إدخال وصلة المؤلف
                newLink = New SingleItemLink
                newLink.LinkType = General.LookupMethods.ByAuthorID
                newLink.LinkID = CInt(dbInfo!ID)
                ItemLinkSet.Add(entryID, newLink)
                ItemLinks.Items.Add(New ListItemData("المؤلف: " & _
                    holdText & " [" & CStr(dbInfo!FullName) & "]", entryID))
            Loop
```

عندما ينقر المستخدم على وصلة متصفح ويب web browser مضمن،فإنه يطلق معالج حدث التصفحNavigating event handler .

```
Private Sub SingleItemDetail_Navigating(ByVal sender As Object, ByVal e As
System.Windows.Forms.WebBrowserNavigatingEventArgs) Handles SingleItemDetail.Navigating
' إلحاق الوصلة التي تم نقرها
        If (e.Url.Scheme = "المكتبة") Then FollowItemLink(CInt(e.Url.Host()))
    End Sub
```

تعود الخاصية e.Url.Scheme بحصة URLقبل الرموز://،بينما تعود e.Url.Host بمكون محدد الشحطة المعكوسة الأول بعد هذا الحروف تماماً.وهو المكان الذي خزنا فيه الفهرس إلى القاموس ItemLinkSet.تستخرج الطريقة FollowItemLink تفاصيل البحث من ItemLinkSet،وتستدعي طريقتنا الموثوقةPerformLookup، منتجةً بحث جديد يتم تخزينه في تاريخ البحث.

```
Private Sub FollowItemLink(ByVal entryID As Integer)
' تقديم موضع حرف في لوحة نص بند مفرد,إتباع الوصلة المشار إليها بذلك البند
        Dim scanLink As SingleItemLink
' إمكانية الوصول إلى الوصلة
        scanLink = ItemLinkSet.Item(entryID)
        If (scanLink Is Nothing) Then Return
' عمل بحث عند الطلب
        If (PerformLookup(scanLink.LinkType, CStr(scanLink.LinkID), False) = False) _
            Then Return
' تخزين التاريخ
        AddLookupHistory(scanLink.LinkType, CStr(scanLink.LinkID))
    End Sub
```

تمكين ميزات البحث.Enabling the Search Features

الفورم ItemLookup جاهزة للاستخدام،نحتاج فقط استدعاءها من حقول البحث على الفورم الرئيسية.اللوحة PanelLibraryItemفي الفورم الرئيسية تتضمن عدة أدوات اختيار صندوق مركبComboBox،ولكن لايوجد كود لملئها،لنعمل على إضافة الكود الآن.انتقل إلى كود الفورم الرئيسية *MainForm.vb* وابحث عن حدث تحميل الفورم MainForm_Load وفيه حتى الآن بعض الكود والذي يعمل على تعديل عناصر الفورم،اعمل على إلحاق الكود التالي إلى نهاية هذا الروتين ليصبح كامل هذا الروتين كما يلي:

```
Private Sub MainForm_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles MyBase.Load
' تحضير الفورم
        Dim sqlText As String
        Dim dbInfo As SqlClient.SqlDataReader
        On Error GoTo ErrorHandler
' ضبط مواقع اللوحات المتنوعة
        PanelLibraryItem.Location = New Point( _
            TaskListBackground.Width, WelcomeBackground.Height)
        PanelPatronRecord.Location = PanelLibraryItem.Location
        PanelHelp.Location = PanelLibraryItem.Location
        PanelCheckOut.Location = PanelLibraryItem.Location
        PanelCheckIn.Location = PanelLibraryItem.Location
        PanelAdmin.Location = PanelLibraryItem.Location
        PanelProcess.Location = PanelLibraryItem.Location
        PanelReports.Location = PanelLibraryItem.Location
        PanelLibraryItem.Visible = True
        ActSearchLimits.PerformClick()
        ' تحميل قائمة نوع البحث
        SearchType.Items.Add(New ListItemData("البحث بالعنوان", LookupMethods.ByTitle))
        SearchType.SelectedIndex = 0
        SearchType.Items.Add(New ListItemData("البحث بواسطة المؤلف", LookupMethods.ByAuthor))
        SearchType.Items.Add(New ListItemData("Lookup By Subject", _
            LookupMethods.BySubject))
        SearchType.Items.Add(New ListItemData("البحث بواسطة الكلمة المفتاحية أي كلمة",
LookupMethods.ByKeywordAny))
        SearchType.Items.Add(New ListItemData("البحث بواسطة الكلمات المفتاحية الجميع",
LookupMethods.ByKeywordAll))
        SearchType.Items.Add(New ListItemData("البحث بواسطة المؤلف", LookupMethods.ByPublisher))
        SearchType.Items.Add(New ListItemData("البحث بواسطة اسم السلسلة", LookupMethods.BySeries))
        SearchType.Items.Add(New ListItemData("البحث بواسطة كود التعريف", _
```

```
            LookupMethods.ByBarcode))
' تحميل أنواع الوسيطات
        SearchMediaType.Items.Add(New ListItemData("<جميع أنواع الوسيطات>", -1))
        SearchMediaType.SelectedIndex = 0
        sqlText = "SELECT ID, FullName FROM CodeMediaType ORDER BY FullName"
        dbInfo = CreateReader(sqlText)
        Do While dbInfo.Read
            SearchMediaType.Items.Add(New ListItemData(CStr(dbInfo!FullName), CInt(dbInfo!ID)))
        Loop
        dbInfo.Close()
' تحميل المواضع
        SearchLocation.Items.Add(New ListItemData("<جميع المواضع>", -1))
        SearchLocation.SelectedIndex = 0
        sqlText = "SELECT ID, FullName FROM CodeLocation ORDER BY FullName"
        dbInfo = CreateReader(sqlText)
        Do While dbInfo.Read
            SearchLocation.Items.Add(New ListItemData(CStr(dbInfo!FullName), CInt(dbInfo!ID)))
        Loop
        dbInfo.Close()
' التحضير للمستخدم كزبون
        UpdateDisplayForUser()
        Return

ErrorHandler:
        GeneralError("MainForm.MainForm_Load", Err.GetException())
        Resume Next
    End Sub
```

يعمل الزر تنظيف على لوحة البحث إعادة إعداد جميع الحقول ويحضرها من أجل بحث جديد،أضف معالج الحدث ActSearchClear_Click كما يلي:

```
Private Sub ActSearchClear_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
ActSearchClear.Click
' إزالة معيار البحث الحالي
        SearchType.SelectedIndex = SearchType.Items.IndexOf(CInt(LookupMethods.ByTitle))
        SearchText.Text = ""
        SearchMediaType.SelectedIndex = SearchMediaType.Items.IndexOf(-1)
        SearchLocation.SelectedIndex = SearchLocation.Items.IndexOf(-1)
    End Sub
```

بما أن مشروع المكتبة من المحتمل أن يتم استخدامه من قبل العديد من الزبائن المختلفين على طول اليوم،سنفرض أن شخص أخر يستخدم البرنامج كل مرة تعود الفورم إلى لوحة البحث.لنحاكي النقر على الزر تنظيف كلما عرض المستخدم اللوحة بحث.اعمل على إيجاد الطريقة TaskLibraryItem،وأضف الكود التالي إلى نهاية الروتين،قبل العبارة SearchText.Focus( ).

```
'التجهيز لبحث جديد
ActSearchClear.PerformClick()
If (ActSearchLimits.Top = LabelMoreLimitsTop.Top) Then ActSearchLimits.PerformClick()
```

مع الاهتمام قدر الإمكان بأن يصبح التطبيق قريب من المستخدم،لنعمل على إضافة "نص التعليمات" إلى لوحة البحث والذي يتنوع بالاعتماد على نوع البحث المختار في القائمة المنسدلة.أضف معالج حدث SearchType_SelectedIndexChanged جديد ومن ثم أضف له الكود التالي:

```
Private Sub SearchType_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles
SearchType.SelectedIndexChanged
        ' البحث تلميحات تحديث
        Select Case CInt(CType(SearchType.SelectedItem, ListItemData))
            Case LookupMethods.ByTitle
                LabelSearchHintsData.Text = " أدخل عنوان أو جزء عنوان في حقل نص البحث  " & _
                    "يمكنك أن تضمن حرف النجمة You may include the asterisk character ( * ) " & _
                    "شاملة كقيمة كحرف"
            Case LookupMethods.ByAuthor
                LabelSearchHintsData.Text = "أدخل اسم المؤلف أو جزء منه " & _
                    "في حقل نص البحث يمكنك تضمين رمز النجمة كقيمة شاملة  " & _
                    "أدخل الاسم في الاسم الأخير,الأول,مفصولة بواسطة الفاصلة  " & "يمكنك أن تحذف الاسم الأخير أو الأول
إذا أردت  "
            Case LookupMethods.BySubject
                LabelSearchHintsData.Text = "أدخل الموضوع أو جزء الموضوع في حقل نص البحث" & _
                    "يمكنك أن تضمن رمز النجمة كقيمة شاملة "
            Case LookupMethods.ByKeywordAny
                LabelSearchHintsData.Text = "أدخل كلمة مفتاحية واحدة أو أكثر في حقل نص البحث " & _
                    "ولكن افصل بينها بفاصلة.لاحظ حروف القمية    " & _
                    "الشاملة غير مسموح بها في بحث الكلمات المفتاحية  " & "الكلمات من أي ستطابق التي البنود
إعادها سيتم المفتاحية"
            Case LookupMethods.ByKeywordAll
                LabelSearchHintsData.Text = "أدخل كلمة مفتاحية واحدة أو أكثر في حقل نص البحث " & _
                    "ولكن افصل بينها بفاصلة.لاحظ حروف القمية" & _
                    "الشاملةc غير مسموح بها في بحث الكلمات المفتاحية ." & _
                    "فقط البنود التي تطابق جميع الكلمات المفتاحية سيتم إعادتها!"
            Case LookupMethods.ByPublisher
                LabelSearchHintsData.Text = "أدخل اسم الناشر أو جزء منه في حقل نص البحث" & _
                    "يمكنك أن تضمن حرف النجمة كحرف قيمة شاملة "
            Case LookupMethods.BySeries
```

```
                LabelSearchHintsData.Text = "أدخل اسم السلسلة أو جزء منها في حقل نص البحث " & _
                    "شاملة قيمة كحرف النجمة حرف تضمن أن يمكنك"
            Case LookupMethods.ByBarcode
                LabelSearchHintsData.Text = "أدخل كود التعريف الرقمي لبند المكتبة في حقل نص البحث" & _
                    "لاحظ أن حرف القيمة الشاملة غير مسموح به في بحث كود التعريف "
            Case LookupMethods.ByDatabaseID
                LabelSearchHintsData.Text = "أدخل المعرف الرقمي لبند المكتبة في حقل نص البحث.لاحظ أن " & _
                    "حروف القيمة الشاملة غير مسموح بها في بحث معرف قاعدة البيانات "
        End Select
    End Sub
```

الشيء الوحيد الباقي،هو عندما ينقر المستخدم على زر بحث،أضف معالج حدث ActSearch_Click ومن ثم أضف كوده كما يلي:

```
Private Sub ActSearch_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles ActSearch.Click
        ' يريد المستخدم أن يعمل بحث على بعض البنود
        Dim mediaLimit As Integer
        Dim locationLimit As Integer
        Dim searchMethod As Integer
        ' تأكد من أن المستخدم قدم معيار البحث المناسب أو الصحيح
        searchMethod = CInt(CType(SearchType.SelectedItem, ListItemData))
        Select Case searchMethod
            Case LookupMethods.ByTitle, LookupMethods.BySubject, _
                    LookupMethods.ByPublisher, LookupMethods.BySeries
                ' نص البحث مطلوب
                If (Trim(SearchText.Text) = "") Then
                    MsgBox("من فضلك اكتب نص البحث", MsgBoxStyle.OkOnly Or MsgBoxStyle.Exclamation,
ProgramTitle)
                    SearchText.Focus()
                    Return
                End If
            Case LookupMethods.ByAuthor
                ' نص البحث مع فاصلة واحدة على الأكثر إذا تطلب الأمر
                If (CountSubStr(SearchText.Text, ",") >= 2) Or _
                        (Trim(SearchText.Text) = "") Then
                    MsgBox("من فضلك اكتب نص البحث مع فاصلة واحدة على الأكثر", MsgBoxStyle.OkOnly Or
MsgBoxStyle.Exclamation, ProgramTitle)
                    SearchText.Focus()
                    Return
                End If
                ' التأكد من أن اسم واحد على الأقل تم كتابته
                If (Trim(GetSubStr(SearchText.Text, ",", 1)) = "") And _
                        (Trim(GetSubStr(SearchText.Text, ",", 2)) = "") Then
                    MsgBox("من فضلك اكتب على الأقل مكون اسم واحد", _
                        MsgBoxStyle.OkOnly Or MsgBoxStyle.Exclamation, ProgramTitle)
                    SearchText.Focus()
                    Return
                End If
            Case LookupMethods.ByKeywordAny, LookupMethods.ByKeywordAll
                ' نص البحث مطلوب
                If (Trim(SearchText.Text) = "") Then
                    MsgBox("من فضلك اكتب نص البحث", MsgBoxStyle.OkOnly Or MsgBoxStyle.Exclamation,
ProgramTitle)
                    SearchText.Focus()
                    Return
                End If
                ' التأكد من عدم تضمين قيم شاملة
                If (InStr(SearchText.Text, "*") > 0) Then
                    MsgBox("رموز القيمة الشاملة أي النجمة غير مسموح بها " & "في بحث الكلمة المفتاحية",
MsgBoxStyle.OkOnly Or _
                        MsgBoxStyle.Exclamation, ProgramTitle)
                    SearchText.Focus()
                    Return
                End If
            Case LookupMethods.ByBarcode, LookupMethods.ByDatabaseID
                ' نص بحث عددي مطلوب
                If (DigitsOnly(SearchText.Text) <> Trim(SearchText.Text)) Or _
                        (Trim(SearchText.Text) = "") Then
                    MsgBox("من فضلك اكتب نص البحث العددي", MsgBoxStyle.OkOnly Or _
                        MsgBoxStyle.Exclamation, ProgramTitle)
                    SearchText.Focus()
                    Return
                End If
        End Select
        ' الحصول على المعايير الموسعة عند الحاجة
```

```
        mediaLimit = -1
        locationLimit = -1
        If (SearchMediaType.Visible = True) Then
            mediaLimit = CInt(CType(SearchMediaType.SelectedItem, ListItemData))
            locationLimit = CInt(CType(SearchLocation.SelectedItem, ListItemData))
        End If
' التمهيد للبحث
        Call (New ItemLookup).InitiateSearch( _
            CType(searchMethod, Library.LookupMethods), _
            Trim(SearchText.Text), mediaLimit, locationLimit)
    End Sub
```

معظم هذا الروتين يعمل على التحقق من المدخلات قبل استدعاء النموذج ItemLookupمن خلال الطريقة InitiateSearch.

```
Call (New ItemLookup).InitiateSearch( _
            CType(searchMethod, Library.LookupMethods), _
            Trim(SearchText.Text), mediaLimit, locationLimit
```

والآن البرنامج جاهز للتشغيل،اعمل على تجريبه الآن.

واجهة المستخدم.User Interface

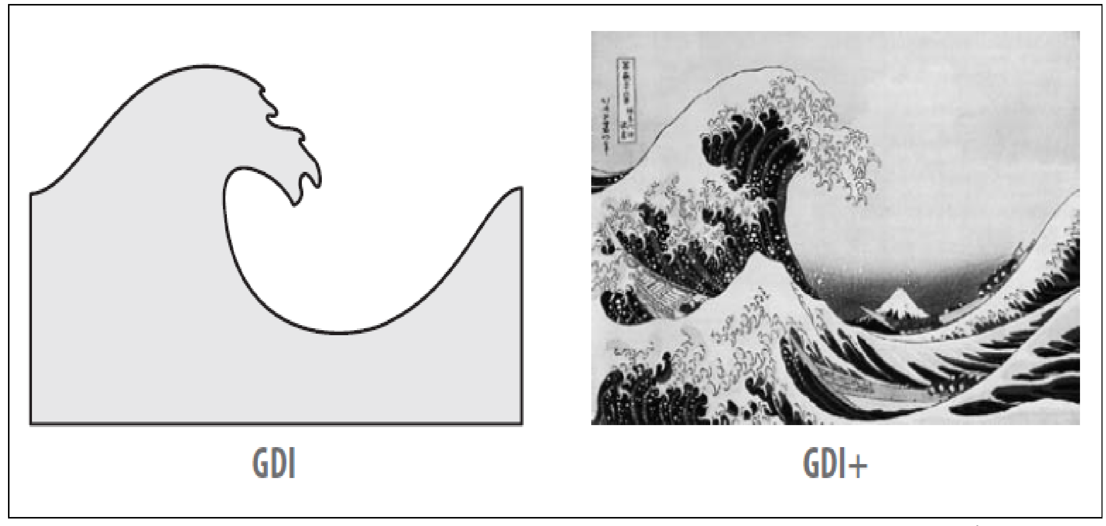
تساوي الصورة ألف كلمة-أو عدة آلاف سطر من الكود المصدري،إذا كنت تعمل على إنتاج صورة نقطية منه.كتابة كود لمعالجة الصور أعماق ألوان متنوعة،أو لرسم طريقة كمية متجهة متعددة الطبقات،يمكن أن يصبح كابوس هندسة الطيات(التشويهات contortions)والجبر الخطيlinear algebra .مما يجعل أحدنا يتوق إلى الأيام السابقة للكمبيوترات ذات شاشات العرض.الفئة البرمجية الأولى التي أخذتها تستخدم *DECWriter*،طابعة معتمدة على نهاية طرفية(موصل طرفي)ليس لديه شاشة، وتتضمن إمكانيات تصويرية ضعيفة.كانت مثالية بالنسبة لي.لا أستطيع رسم خط مباشر بأي طريقة.

لحسن الحظ من أجل هواة الفن في كل مكان،فقد قطعت الكمبيوترات شوطاً طويلاً في دائرة الرسوميات. *GDI+*نظام الرسم القياسي للدوت نت،يتضمن ميزات رسم معقدة والتي ستجعل DECWriter يبكي.وهو مبني على تقنية ويندوز أقدم"واجهة أداة الرسمGraphics Device Interface "(GDI)،تتضمن أوامر لرسم خطوط lines،نصوصtext ،وصور imagesثنائية البعد2D.خلفGDI+ ،توفر الدوت نت أيضاً دعم لأساسيات عرض ويندوز Windows Presentation Foundation (WPF) الأحدث،واجهة المستخدم الغنية rich user interface ونظام التقديم(العرض) المتعدد الوسائط multimedia presentation system والذي يعتمد بشكل جزئي على XML.تتضمن أساسيات تقديم (عرض)ويندوز WPFالعرض والتفاعل مع الميزات التي تذهب أبعد منGDI+ ، ويوجد أيضاً عدة ميزات GDI+غائبة عن WPF.على الرغم من أنني سأعطي لمحة عامة عن WPF في هذا الفصل،معظم الفصل (وجميع كود واجهة المستخدم لمشروع المكتبة)سيركز على GDI+ .

نظرة شاملة علىGDI+ . Overview of GDI+.

قبل الدوت نت كان مبرمجي ويندوز يعتمدون على نظام GDIلرسم أي شيء تقريباً على الشاشة،حتى ولو كانوا لايعرفوا بوجود GDI.بالإضافة إلى الصور النقطية bitmap images، جميع الأدواتcontrols،اللصاقاتlabels، حواف النافذةwindow borders، والأيقونات iconsالتي تظهر على الشاشة كل هذا بفضل GDI. كانت GDI خطوة عملاقة للأمام من الرسوم الحرفية(الرمزية).قدمت GDI مجموعة قاعدية من ميزات الرسم والتي منها تستطيع بصورة كامنة إخراج أي نوع من الصور المعقدة.ولكن لم تكن سهلة.كانت الرسوم بدائية،وكان عليك أن تنمي build up أنظمة معقدة من الجزئيات. معظم المبرمجين لم يهتموا بعمل الأشياء جميلة،لذلك فإنهم يحاولون تجنب تعقيدات GDI.ولكن في بعض الأحيان يكون عليك رسم خط أو دائرة،ولم تكن هناك طريقة بخصوص ذلك.

إنGDI+ جديدة في الدوت نت،مبنية علىGDI، وتعمل على توفير البدائيات الأساسية basic primitives لـ GDI.ولكن تعمل أيضاً على تزويد بعض التجمعات الأكثر تعقيداً لميزات الرسوم في دوال سهلة الاستخدام.هذه البساطة أدت إلى النهضة الأوربية renaissanceلمبرمجي الأعمال الرسومية الأولية.انظر إلى الشكل التالي، والذي يبين صورة كانت قد رسمت باستخدام GDI،ونفس الصورة الناتجة فقط بعدة أوامر سريعة في GDI.



يجعل نظام GDI+موطنه فضاء الأسماءSystem.Drawing، ويتضمن حشود multitudesمن الفئات التي تمثل كائنات الرسمdrawing objects، السطوح surfaces،وميزات الزخرفة embellishment التي تعمل على تمكين عرض الرسوميات.ولكن الأمر لايتعلق فقط بالعرض.تستقرئ GDI+الرسوم المتجهة vector والنقطية bitmapعلى جميع سطوح المخرجات المتاحة:الرسومات النقطية bitmapsأو الخطية line على الشاشة(متضمنةً أسطح الفورم والأدوات)، مخرجات تقرير report output على الطابعة،الزخرفات(النقوشgraffiti)على خلفية جدار السوبر ماركت خاصتك،محتوى الصور المعدة من أجل ملف JPEG -جميعها تنتمي إلى نفس GDI+.جميع المقاصد تستخدم نفس طرق methodsوكائنات objects الرسم،وتجعلها أسهل لك في استقراء كود الرسم خاصتك.

تتضمن ميزات الGDI+ السطوحsurfaces، حبر الرسمdrawing inks، عناصر الرسمdrawing elements، والتحويلاتtransformations .

- تستقرئ الـ GDI+سطوح الرسم drawing *surfaces* من خلال الفئة System.Drawing.Graphics.

يمثل هذا الكائن لوحة(نسيج) الرسم drawing canvas،مع مواصفات من أجل عمق الألوان والحجم(العرض والارتفاع).يمكن أن تكون اللوحة على صلة بالإعدادات الإقليمية لشاشة الكمبيوتر أو الشبكة المحلية،المساحة المحفوظة الداخلية من أجل المخرجات النهائية إلى الطابعة،أو لوحة رسومية عامة لمعالجة محتوى في الذاكرة قبل إخراجه إلى العرض أو إلى ملف.نوع أخر للسطح surface ، وهو المسار *path* (System.Drawing.Drawing2D.GraphicsPath)مشابه للمسجل الماكروي macro recorderمن أجل الرسوميات المتجهة vector (خطوط line).الرسم المعمول ضمن مسار path يمكن أن يتم "إعادة تشغيلهreplayed " على سطح رسم قياسي،أو يتم استخدامه لتوفير حدود من أجل أوامر الرسم الأخرى.

- الألوان والحبر Colors and inks تظهر في نموذج الألوان *colors* (قيم لون كامدة أو شبه شفافة)،ألفراشي *brushes* (bitmapالنقطية المعتمدة على شبه قلم pseudo-pensالمستخدم من أجل التعبئة fillsوالإكساء(الطلاءtiling)) والأقلام *pens* (كائنات رسم خط ملون colored line-drawing objects بسماكة معينة)

•تتضمن عناصر الرسم Drawing elements المستطيلاتrectangles، القطوع الناقصةellipses، الخطوطlines،وأشكال أخرى قياسية أو مخصصة.وهي تتضمن أيضاً الخطوط،وكل من النسخ المعتمدة على الخطوط(الحدود) الخارجية والمرتبطة بالنقاط(البت).

.تتيح لك التحويلات *Transformations* إعادة تحجيمresize، تدويرrotate، وحرف(تشويه skew) الرسوميات عندما تعمل على إنتاجها.عندما يتم تطبيق تحويل على سطح،تستطيع رسم كائنات وكأنه لم يتم تطبيق تحويل عليها،وستحدث التغيرات في الوقت الحقيقي.

أدوات نماذج ويندوز التي تستخدمها في تطبيقات سطح المكتب تحترس بشكل عام من ميزات العرض الخاصة بها.مهما يكن ،بعض الأدوات تتيح لك تولي بعض أو كل مسؤوليات الرسم.على سبيل المثال،أداة صندوق القائمة ListBoxتعرض نص بلون مفرد بسيط من أجل بند قائمة.مهما يكن،تستطيع(إعادة قيادة) أو الهيمنة على رسم كل بند قائمة،موفراً محتوى خاص بك،والذي من المحتمل أن يحتوي نص أو رسوميات متعددة الألوان.هذه المقدرة على تزويد بعض من كود الرسم إلى أداة يعرف حامل(صاحب)الرسم*owner draw*، وهو يعمل من خلال نفس كائن الرسم المعمم Graphicsالمستخدم من أجل الرسوميات الأخرى.سنعمل على تضمين بعض كود الرسم الخاص بنا في مشروع المكتبة.

هذا الفصل لن يغطي كامل ميزات +GDIالمتاحة فهو لن يغطي أكثر من 1% منها،حيث أن +GDIمعقد وضخم،ويمكن أن تمضي سنوات وأنت تفتش ضمن كل ميزة صغيرة،ولمزيد من المعلومات راجع مستندات MSDN.

اختيار لوحة رسم Selecting a Canvas

معظم الرسم في الدوت نت يحدث في سياق كائن الرسومياتGraphics object (بالنسبة لأولئك الذين على إطلاع بالتطوير السابق للدوت نت في ويندوز،هذا مشابه لسياق *device*).توفر كائنات الرسوميات لوحة رسم canvasوالتي ترسم بها الخطوط،الأشكال،الصور النقطيةbitmap،و مختصرات (الماكرو)الرسم المسجلة مسبقاً prerecorded drawing macros.كائنات الرسوميات لاتحتوي على سطح رسومي بنفسها،فهي موصلات (قنواتconduits) شاملة إلى اللوحات الحقيقية.

يوجد دائماً سطح ما خلف كائنات الرسوميات،سواء كانت جزء من الشاشة،كائن نقطي ما،أو سطح المحاكاة لصفحة مطبوعة.أي رسم يتم عمله إلى كائن الرسوميات يتأثر مباشرةً بالسطح المضمن.يتضمن كائن الرسوميات الكثير من الطرق التي تتيح لك رسم أشكال وصور على سطح الرسوميات،وتنجز نشاطات سحرية أخرى ثنائية البعد2D. وسنغطي العديد منها في هذا الفصل.

إنشاء والحصول على كائنات رسومية Obtaining and Creating Graphics Objects.

إن الحصول على كائن رسومي من أجل نموذج على الشاشة أو أداة سهل بقدر سهولة استدعاء طريقة أداة النموذج CreateGraphics.

```
Dim wholeFormGraphics As Graphics = Me.CreateGraphics()
Dim buttonOnlyGraphics As Graphics = Button1.CreateGraphics()
```

بعض الأحداث،والأكثر ملاحظةً حدث الطلاء Paint eventمن أجل الأدوات والنماذج،توفر إمكانية الوصول لكائن الرسوميات من خلال المعاملات النسبية للحدثevent arguments .

```
Private Sub PictureBox1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles PictureBox1.Paint
        Dim paintCanvas As Graphics = e.Graphics
    End Sub
```

تستطيع أيضاً إنشاء كائن رسوميات غير متعلق بأي منطقة عرض موجودة وذلك بإرفاقه إلى شبكة نقطية bitmap .

```
Dim trueBitmap As New Bitmap(50, 50)
Dim canvas = Graphics.FromImage(trueBitmap)
```

تذكر ،جميع التغيرات المعمولة على حالة canvas ستؤثر على الصورة trueBitmap.

التخلص من كائنات الرسوميات بصورة صحيحة Disposing of Graphics Objects Properly.

عندما تنتهي من العمل مع كائن الغرافيك(الرسوميات)الذي عملت على إنشاءه،يجب عليك التخلص منه باستدعاء الطريقةDispose. (هذه القاعدة صحيحة من أجل العديد من كائنات +GDIالمختلفة).يجب ،ويجب ،ويجب عليك التخلص منه عندما تنهي العمل معه.فإذا لم تفعل،فقد يسبب فساد في الصورة،قضايا استخدام الذاكرة،أو حتى الأسوأ،لذلك ومن فضلك تخلص من جميع كائنات الغرافيك تماماً.

```
canvas.Dispose()
```

إذا أنشأت كائن غرافيك ضمن حدث،ستحتاج في الحقيقة إلى التخلص منه قبل الخروج من معالج الحدث ذاك.ولا توجد ضمانة لبقاء كائن الغرافيك محقق في حدث لاحق. بالإضافة،إلى أنه من السهل إعادة إنشاء كائن غرافيك آخر في أي وقت.

إذا كنت تستخدم كائن غرافيك تم تمريره لك من خلال جزء آخر من البرنامج(مثل المرجع e.Graphics في معالج حدث الطلاء السابق)،ليس عليك التخلص منه.كل منشئ مسئول عن التخلص من كائناته الخاصة به(التي يملكها).

اختيار الأقلام و الفراشي Choosing Pens and Brushes.

تتضمن الكثير من أعمال الرسوميات بدائيات الرسم:استخدام الخطوط،القطوع ،المستطيلات،وأشكال أخرى نظامية أو غير نظامية لتركيب العرض النهائي.وكما في الحياة الحقيقية ،إنك ترسم هذه البدائيات باستخدام كائن قلمPen object .من أجل تلك البدائيات التي تنتج في شكل قابل للتعبئة أو شبه قابل للتعبئة،يعين كائن الفرشاة اللون أو النموذج للاستخدام في منطقة التعبئة تلك.يتضمن الـ +GDIالعديد من الأقلام pensوالفراشي brushes المعرفة مسبقاً،أو بإمكانك إنشاءها بشكل خاص بك.

الأقلام.Pens

الأقلام هي أدوات رسم الخطوط تستخدم مع أوامر الرسم لكائن الغرافيك.القلم الأساسي لديه لون نقي(صافي)solid color وسماكةthickness .

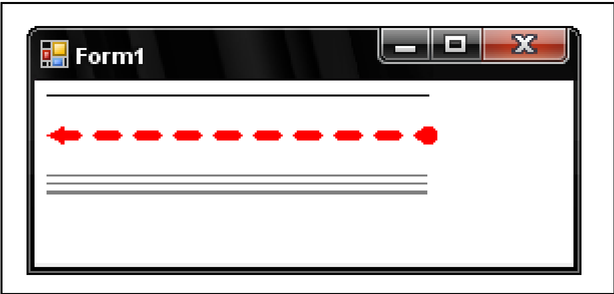
```
'قلم أحمر بعرض خمس وحدات
Dim redPen As New Pen(Color.Red, 5)
```

وكما مع كائنات الغرافيك،أي قلم تعمل على إنشاءه يستخدم الكلمة المحجوزة Newيجب أن يتم التخلص منه بشكل دقيق عند الانتهاء من استخدامه.

```
redPen.Dispose()
```

العديد من الأقلام المسبقة التعريف تم جعلها متاحة من خلال الفئة System.Drawing.Pens،جميعها مسماة بألوانها،كما فيPens.Red ،إذا استخدمت واحد من هذه الأقلام،ليس عليك التخلص منه.
تستطيع إنشاء الكثير من الأقلام الهامة والتي تتنوع بأنماط الخط،ديكورات النهاية،وتنوعات اللون.الكود التالي يولد الصورة المعروضة في الشكل التالي:

```
    Private Sub PictureBox1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs)Handles PictureBox1.Paint
        ' رسم بعض الخطوط الجميلة
        Dim usePen As Pen
        ' تفريغ الخلفية
        e.Graphics.Clear(Color.White)
        ' أرسم خط أساسي بسماكة 1بكسل يستخدم لون شريط العنوان
        usePen = New Pen(SystemColors.ActiveCaption, 1)
        e.Graphics.DrawLine(usePen, 10, 10, 200, 10)
        usePen.Dispose()
        ' ارسم خط متقطع احمر مع سهم وكرة في النهايات.كل مقطع تقطيع لديه نهاية مدببة
        usePen = New Pen(Color.FromName("Red"), 5)
        usePen.DashCap = Drawing2D.DashCap.Triangle
        usePen.StartCap = Drawing2D.LineCap.ArrowAnchor
        usePen.EndCap = Drawing2D.LineCap.RoundAnchor
        usePen.DashStyle = Drawing2D.DashStyle.Dash
        e.Graphics.DrawLine(usePen, 10, 30, 200, 30)
        usePen.Dispose()
        ' قلم أسود شبه شفاف مع خط بثلاث أجزاء,إثنان رفيعان وواحد سميك.
        usePen = New Pen(Color.FromArgb(128, 0, 0, 0), 10)
        usePen.CompoundArray = _
        New Single() {0.0, 0.1, 0.4, 0.5, 0.8, 1.0}
        e.Graphics.DrawLine(usePen, 10, 55, 200, 55)
        usePen.Dispose()
    End Sub
```



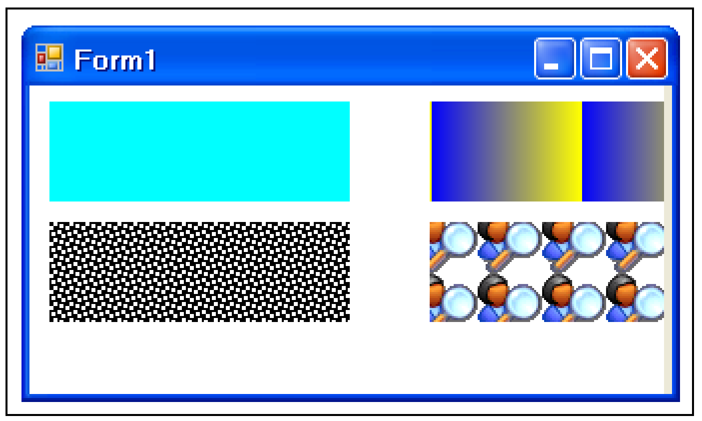
يبين الكود أن هناك عدة طرق مختلفة لتعين اللون،إما بواسطة الاسم المعرف مسبقاً(Color.White وSystemColors.ActiveCaption)،و كاسم نصي(باستخدام Color.FromName)،أو بواسطة قيم حروفه الأحمر-الأخضر-الأزرق Alpha-Red-Green-Blue (Color.FromArgb). تلك النسخة الأخيرة تتيح لك تزويد قيم متميزة لـ "alpha blend"(والتي تعمل على وضع مستوى الشفافية،من 0من أجل شفافية كاملة إلى255كامد بالكامل(عدم شفافية))الأحمر،الأخضر والأزرق هي المكونات بالنسبة لكامل الألوان.
معظم الخاصيات المعينة لقلم والتي وضحتها هنا نوعاً ما واضحة وتشرح نفسها.وكما مع معظم الـGDI+ ، كمية الميزات المتاحة والتي تخدر العقل تجعل من المستحيل توثيقها كاملاً في فصل واحد صغير،بكل بساطة سأحيلك إلى مستندات النت من أجل فئة القلم Penللحصول على التفاصيل البراقة.

الفراشي.Brushes

تستخدم الفراشي من أجل ملئ الفراغات بين خطوط مرسومة،حتى لو جعلت تلك الخطوط بالكامل غير مرئية.تتضمن الـ GDI+تشكيلة لأنواع ألفراشي،من ضمنها ألفراشي الجامدة(النقيةsolid brushes)(فرشاتك الأساسية الوحيدة اللون)،فراشي التظليل *hatch brushes* (فراشي نموذج والتي تكون ممتعة ولكنها عامة)، الفراشي البنيوية texture brushes (حيث يتم استخدام نمط نقطي bitmapمن أجل الفرشاة)،وفراشي متدرجةgradient brushes (والتي تخفت fadeببطء من لون واحد إلى أخر عبر(على طول) الفرشاة).تتضمن الفئة System.Drawing.Brushes بعض ألفراشي الصافية(الصلبة) المعرفة مسبقاً والمعتمدة على اسم اللون.وكما مع الأقلام،عليك التخلص من ألفراشي التي قمت بإنشائها،وليست تلك ألفراشي الصلبة المعرفة مسبقاً من قبل النظام.
يرسم المقطع التالي من الكود بعض المستطيلات البسيطة بتشكيلات من أنماط الفراشي،وتظهر النتيجة في الشكل التالي.

```
Private Sub PictureBox1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles PictureBox1.Paint
        ' رسم بعض المستطيلات الجميلة
        Dim useBrush As Brush
        e.Graphics.Clear(Color.White)
        ' رسم مستطيل ملون بلون صافي
        e.Graphics.FillRectangle(Brushes.Cyan, 10, 10, 150, 50)
        ' رسم مستطيل مظلل.استخدم الأسود من أجل الخلفية,والأبيض من واجهة النموذج.
        useBrush = New Drawing2D.HatchBrush( _
        Drawing2D.HatchStyle.LargeConfetti, _
        Color.White, Color.Black)
```

```
        e.Graphics.FillRectangle(useBrush, 10, 70, 150, 50)
        useBrush.Dispose()
'        رسم مستطيل بشكل خطي من اليسار إلى اليمين
'        يحدد تدرج المستطيل بعينه بداية التغير,بالاعتماد على أصل سطح الغرافيك
        useBrush = New Drawing2D.LinearGradientBrush(New Rectangle(200, 10, 75, 25), Color.Blue,
Color.Yellow, Drawing2D.LinearGradientMode.Horizontal)
        e.Graphics.FillRectangle(useBrush, 200, 10, 150, 50)
        useBrush.Dispose()
'        استخدم صورة من أجل الفرشاة.
        useBrush = New TextureBrush(Image.FromFile("C:\LookupItem.bmp"))
        e.Graphics.FillRectangle(useBrush, 200, 70, 150, 50)
        useBrush.Dispose()
    End Sub
```



إلحاق نص من الخط . Flowing Text from the Font

الدوائر والمربعات رائعة،ولكنها في الكثير من الأحيان لا تكون على اتصال دائم.معظمنا يعتمد على نص لنقول ما نعني.لحسن الحظ،الـGDI+ لديها ميزات بوفرة تعمل على وضع نص على سطح الغرافيك(الرسوميات) خاصتك. كل الغضب قبل واجهات المستخدم الرسومية كان منصب على أن النصوص لم تكن القضية الحقيقية،فإما أنك تستخدم الرموز(الحروف)المبنية ضمن النظام،أو لاتستخدم شيء. على الشاشة تم تصميم كل حرف من الأبجدية ضمن مكون صلب للكمبيوتر(هارد وير)أو العارض monitor وأي حرف خاص يمكن أن يظهر فقط ضمن مربع من شبكة معرفة مسبقاً80× 24 .كانت الطابعات أفضل قليلاً،بما أنك تستطيع التراجع خطوة للوراء وتعيد الكتابة(الطابعات) على مواضع مكتوبة(مطبوعة)سابقاً لإنتاج إما نص بخط غامق أو نص بشحطة منخفضة.ومع ذلك،كنت بشكل عام مقيد بخط واحد فقط،أو بمقدار ضئيل من الخطوط الأساسية المغلقة في ذاكرة الطابعة فقط .

مثل هذه القيود أصبحت أشياء من الماضي.جميع النصوص في ميكروسوفت ويندوز تظهر بشكل لائق من الخطوط،رسومات لأشكال الحروف والتي يمكن إعادة تحجيمها،أو تمديدها أو تأكيدها emphasizedلتلاقي احتياجات أي نص.ولأن بإمكان المستخدم إضافة خطوط إلى النظام في أي وقت،ومن أي مصدر آخر،فإن تشكيلة هذه الخطوط تصبح مدهشة.ولكنك تعلم سابقاً كل هذا،لذلك دعنا ندخل إلى الكود.

للحصول على إمكانية الوصول إلى خط ما من أجل استخدامه في رسومياتك،أنشئ حالة من الفئة System.Drawing.Font ومرر لها على الأقل اسم خط ،وحجم النقطة، وبشكل اختياري مرجع التخطيط:

```
Dim basicFont As New Font("Arial", 14, FontStyle.Italic)
```

من الطبيعي،أن تتنوع قائمة الخطوط المتاحة بتنوع النظام،إذا كنت ذاهب لاستخدام خطوط أبعد من تلك الخطوط الأساسية المثبتة مع ويندوز،عليك تأكيد أن الخط المسمى متاح حقاً،ولديه خيار تراجعي إذا لم يكن كذلك.تستطيع الحصول على قائمة بجميع الخطوط بطلبها من GDI+بشكل جميل.جميع الخطوط تظهر في "عائلاتfamilies،" حيث أن كل عائلة مسماة يمكن أن يكون لديها غامقbold، مائلitalic، وتنوعات variationsأخرى مثبتة كملفات خط منفصلة.الكود التالي يضيف قائمة بجميع عائلات الخطوط إلى أداة صندوق قائمةListBox :

```
Dim allFonts As New Drawing.Text.InstalledFontCollection()
        For Each oneFamily As Drawing.FontFamily In allFonts.Families
            ListBox1.Items.Add(oneFamily.Name)
        Next oneFamily
```

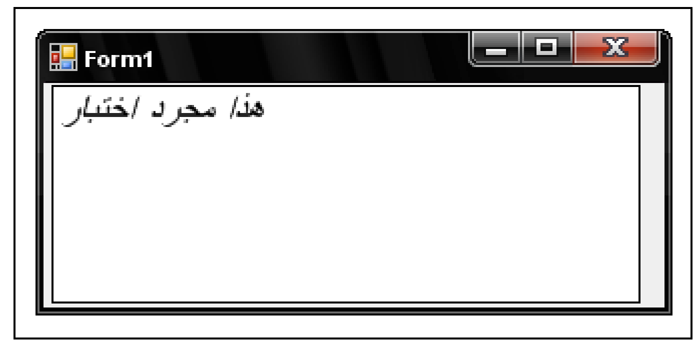
إذا كان الخط الذي تحتاجه غير متاح ولست متأكد من ما ستستخدم،دع GDI+ يختار لك.فهو يتضمن عدة خطوط شاملة للاستخدام الطارئ:

```
Drawing.FontFamily.GenericMonospace
Drawing.FontFamily.GenericSansSerif
Drawing.FontFamily.GenericSerif
```

بالعودة إلى الخطوط المستخدمة فعلياً في الرسم،يتضمن كائن الغرافيك الطريقة DrawStringوالتي تضع بعض النصوص في اللوحة:

```
Private Sub PictureBox1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles PictureBox1.Paint
        Dim basicFont As New Font("Arial", 14, FontStyle.Italic)
        e.Graphics.DrawString("هذا مجرد اختبار", basicFont, Brushes.Black, 0, 0)
        basicFont.Dispose()
    End Sub
```

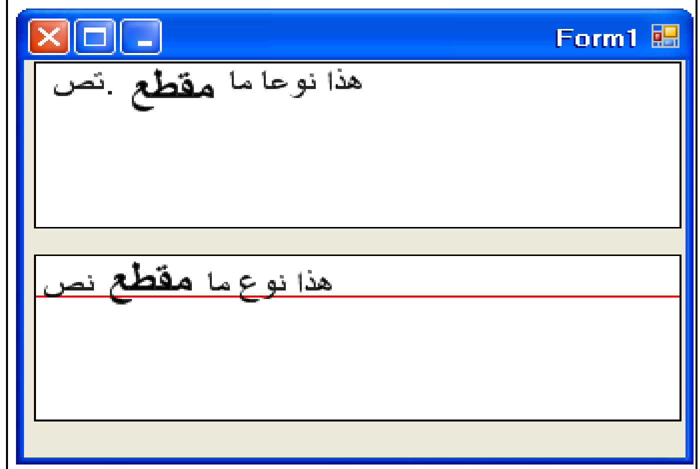
يبين الشكل التالي مخرجات هذا الكود.في معظم عينات الكود لهذا الفصل،سأعمل على إخراج المحتوى إلى أداة صندوق صورة PictureBoxمسماة PictureBox1 والتي عملت على وضعها على الفورم لتطبيق نماذج ويندوز جديد.وعملت على وضع خاصية هذه الأداة BorderStyle إلىFixedSingle،وخاصية BackColor إلى أبيض Whiteبحيث أستطيع إظهار visualize حوافedges اللوحة.يحدث الرسم في معالج حدث التلوينPaint ، والذي يتم استدعاءه عندما يحتاج صندوق القائمة لإعادة تنشيط، عندما تحجبه نافذة أخرى وبالتالي لايظهر.في باقي أمثلة الكود،لن أعمل على تضمين تعريف الروتين Sub PictureBox1_Paint،ما سأبينه لك هو فقط الكود الذي يظهر داخل هذا الروتين.



بالطبع ،تستطيع مزج ومطابقة الخطوط على لوحة مخرجات مفردة.يتضمن هذا الكود نص يستخدم Arial 14وArial 18:

```
Dim basicFont As New Font("Arial", 14)
        Dim strongFont As New Font("Arial", 18, FontStyle.Bold)
        Dim offset As Single = 0.0
        Dim showText As String
        Dim textSize As Drawing.SizeF
        showText = " .نص "
        textSize = e.Graphics.MeasureString(showText, basicFont)
        e.Graphics.DrawString(showText, basicFont, _
        Brushes.Black, offset, 0)
        offset += textSize.Width
        showText = "مقطع"
        textSize = e.Graphics.MeasureString(showText, strongFont)
        e.Graphics.DrawString(showText, strongFont, _
        Brushes.Black, offset, 0)
        offset += textSize.Width
        showText = "هذا نوع ما"
        textSize = e.Graphics.MeasureString(showText, basicFont)
        e.Graphics.DrawString(showText, basicFont, _
        Brushes.Black, offset, 0)
        offset += textSize.Width
        strongFont.Dispose()
        basicFont.Dispose()
```

تظهر مخرجات هذا الكود في الشكل التالي،وذلك في الصندوق العلوي،ولكن إني أريد الحواف الدنيا لأجزاء الجسد الرئيسية بالنسبة لكل مقطع نص- بالخطوط الأساسية لكل مقطع-مرصوفة بشكل لائق،كما هو مبين في الصندوق السفلي.



عمل جميع أشياء ترصيف خط الأسطر المحببة هو نوع من الإزعاج ،عليك عمل جميع أنواع القياسات بالاعتماد على تصميم الخط الأصلي كما يستقرئ extrapolated في البكسل بالاعتماد على جهاز الشاشةpixel-based screen device. ومن ثم تعمل على وصل عظم الركبة إلى عظم الفخذ،وهكذا.إليك الكود الذي يعمل على إنتاج الصورة الثانية للسطر المرصف.

```
Private Sub PictureBox2_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles PictureBox2.Paint
        Dim basicFont As New Font("Arial", 14)
```

```
        Dim strongFont As New Font("Arial", 18, FontStyle.Bold)
        Dim offset As Single = 0.0
        Dim showText As String
        Dim textSize As Drawing.SizeF
        Dim basicTop As Single
        Dim strongTop As Single
        Dim strongFactor As Single
        Dim basicFactor As Single
        ' عائلة الخط تستخدم وحدة التصميم,من الممكن تخصيصها بواسطة المصمم الأصلي للخط
        ' أربط هذه الوحدات إلى وحدات العرض(النقاط).كمايلي
        strongFactor = strongFont.FontFamily.GetLineSpacing( _
        FontStyle.Regular) / strongFont.Height
        basicFactor = basicFont.FontFamily.GetLineSpacing( _
        FontStyle.Regular) / basicFont.Height
        ' حدد موضع كل الخط بالنسبة لكل خط أساسي
        strongTop = (strongFont.FontFamily.GetLineSpacing( _
        FontStyle.Regular) - strongFont.FontFamily.GetCellDescent( _
        FontStyle.Regular)) / strongFactor
        basicTop = (basicFont.FontFamily.GetLineSpacing( _
        FontStyle.Regular) - basicFont.FontFamily.GetCellDescent( _
        FontStyle.Regular)) / basicFactor
        ' ارسم الخط الذي يعمل على تثبيت نص مرصف
        e.Graphics.DrawLine(Pens.Red, 0, strongTop, _
        e.ClipRectangle.Width, strongTop)
        ' أظهر كل جزء من النص
        showText = " .نص "
        textSize = e.Graphics.MeasureString(showText, basicFont)
        e.Graphics.DrawString(showText, basicFont, _
        Brushes.Black, offset, strongTop - basicTop)
        offset += textSize.Width
        showText = "مقطع"
        textSize = e.Graphics.MeasureString(showText, strongFont)
        e.Graphics.DrawString(showText, strongFont, _
        Brushes.Black, offset, 0)
        offset += textSize.Width
        showText = "هذا نوعا ما"
        textSize = e.Graphics.MeasureString(showText, basicFont)
        e.Graphics.DrawString(showText, basicFont, _
        Brushes.Black, offset, strongTop - basicTop)
        offset += textSize.Width
        strongFont.Dispose()
        basicFont.Dispose()
    End Sub
```

تجري حسابات أكثر في الكود، وأنا لم أحاول حتى معالجة tackle الأشياء مثل سد الفراغات بين الأحرفkerning، ربط الحروفligatures، أو أي شيء أخر يجب عمله على المادة المطبوعة.على أية حال،إذا احتجت إلى عمل معالجة معقدة للخط،تعمل +GDIعلى عرض جميع التفاصيل بحيث تستطيع عملها كما ينبغي.إذا كنت تريد فقط إخراج سطر بعد سطر في نص باستخدام نفس الخط،استدعي طريقة الخط GetHeightمن أجل كل سطر معروض:

```
verticalOffset += useFont.GetHeight(e.Graphics)
```

يكفي الآن أشياء معقدة.توجد أشياء سهلة وباردة على القلب يمكن عملها على النص أيضاً.ألم تلاحظ أن مخرجات النص تستخدم الفراشي وليس الأقلام؟وهذا يعني أن بإمكانك رسم النص باستخدام أي فرشاة تستطيع إنشاءها.ومقطع الكود التالي يستخدم فرشاة صورة نقطية bitmap للصورة LookupItem لمشروع المكتبة لعرض نص ما معتمد على صورة نقطية bitmap.ستظهر المخرجات في الشكل الذي يلي الكود مباشرةً.

```
Dim useBrush As Brush = New TextureBrush(Image.FromFile("C:\LookupItem.bmp"))
Dim useFont As New Font("Arial", 60, FontStyle.Bold)
e.Graphics.DrawString("Wow!", useFont, useBrush, 0, 0)
useFont.Dispose()
useBrush.Dispose()
```



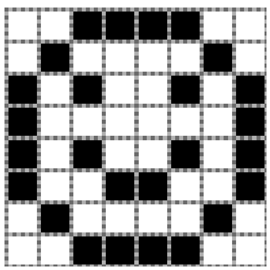
إندماج النص والرسوميات.

تخيل الصور Imagining Images

من المحتمل وأكثر من أي شيء أخر،أن الانترنت قد أججت معدل حاجة مستخدمي الكمبيوتر للمحفزات البصرية .فمواقع الويب مغمورة بـ الجيف، ،وتشكيلة من تنسيقات الصور الأخرى.حتى لو كنت تتعامل مع تطبيقات غير الويب،من المحتمل أنك،وكمبرمج،يحدث لأن تحتكّ وعلى نحو متكرر مع صور رسومية.لحسن الحظ،يتضمن الـ GDI+ميزات تتيح لك إدارة ومعالجة هذه الصور بشكل سهل.
تنسيق الملف "BMP" هو التنسيق للصورة النقطية الأولي(الأصلي) native bitmap format المضمن في ويندوز ميكروسوفت،ولكنه ليس بتلك الأهمية في عالم الويب.ولكن ولا شيء من هذا يشكل مشكلة بالنسبة لـ GDI+.فيمكن تحميل وإدارة الملفات باستخدام تنسيقات الرسوميات(الغرافيك) التالية:

- ملفات صور نقطية "BMP" للويندوز بأي عمق للون وبأي حجم.
- ملفات التنسيق التبادلي للرسومات الخاصة (لخدمة الكمبيوتر)CompuServe Graphics Interchange Format، "الجيفGIF" ، تستخدم بشكل شائع من أجل الصور الغير فوتوغرافية(تصويرية) على الانترنت.
- ملفات مجموعة الخبراء الفوتوغرافية المترابطة Joint Photographic Experts Group، شائعة الاستخدام من أجل الصور الضوئية photos والصور الأخرىimages على الانترنت.ملفات JPEG تكون مضغوطة داخلياً compressed internally لتخفيض حجم الملف،ولكن مع احتمال فقدان نوعية الصورة. ملفات ملف الصورة القابلة للاستبدال(“EXIF”) Exchangeable Image File، تنوع عن JPEG والتي تخزن صور فوتوغرافية(تصويرية)photographs) احترافية.
- ملفات الرسوم الشبكية القابلة للنقل(المحمولة)(“PNG”) Portable Network Graphics)،والتي هي مشابهة لـملفات GIF،ولكن مع بعض الميزات المحسنة.
- ملفات تنسيق ملف صوري رقعي(تجميعي)(“TIFF”) Tag Image File Format، والتي هي نوع من ضم (دمج)جميع تنسيقات الملفات الأخرى.بعض التنظيمات(المنظمات) الحكومية تخزن الصور الممسوحة(المدققة) باستخدام TIFF.
- ملفات توصيفيهMetafiles، والتي تخزن مبادئ خط الكمية المتجهة vector line art بدلاً من الصور النقطية.
- ملفات الأيقونة (“ICO”) Icon files)،والتي يتم استخدامها من أجل أيقونات ميكروسوفت ويندوز القياسية.تستطيع تحميلها كصور نقطيةbitmaps، ولكن يوجد أيضاً الفئة Icon المميزة والتي تتيح لك معاملتها بطرق مشابهة أكثر للأيقونات.

ثلاث فئات رئيسية يتم استخدامها من أجل الصور:الصورة Image(فئة قاعدية مجردة من أجل كل من الفئتين الأخريين).،النقطية Bitmap،وملف التوصيفيMetafile . سأناقش فئة الميتافايل فيما بعد.
تمثل البيتماب Bitmaps (الصور النقطية) صورة image مثلما تم رسمها على شبكة من البتات(الوحدات التخزينية).عندما يكون بت على شبكة فعالـ*on*، فإن خلية الشبكة تلك تكون مرئية أو معبئة. وعندما يكون البت معطلـ*off*، فإن خلية الشبكة تكون غير مرئية أو فارغة.يبين الشكل التالي صورة بسيطة باستخدام الشبكة النقطية(شبكة البتات).



بما أن البت بإمكانه دعم حالتين فقط،فإن ملفات صورة نقطية بت واحد1-bit bitmap files تكون صورة أحادية اللون monochrome، تعرض الصور باستخدام فقط اللون الأسود والأبيض.لتضمين ألوان أكثر،تضيف الصور النقطية bitmapsمستويات أو سطوح planes إضافية .يتم تنضيد stackedالمستويات planes فوق بعضها البعض لذلك فإن خلية في مستوى واحد تتطابق مع تلك الخلية التي بنفس الموضع في جميع المستويات الأخرى.فمجموعة من 8مستويات تنتج في "صورة نقطية بـ8 بت8-bit bitmap image"،وتدعم 256 لون في كل خلية(لأن $2^{مستوى}=2^8=256$). تتضمن بعض الصور بحدود 32 أو حتى64 بت bits (مستوى planes)،وأيضاً بعض من هذه المستويات(البتات) يمكن أن يتم حجزها من أجل" مزيج(أشعة) ألفاalpha blending" والتي تجعل ملاحظة(إدراكperceived) الشفافية transparencyممكنة.

إن معالجة كل هذه البتات واجب مزعج،لحسن الحظ،ليس عليك القلق بالنسبة لها،بما أنه يتم عمل كل هذا لك بواسطة الفئة Bitmap.تحتاج فقط إلى تحميل وحفظ الصور النقطية(باستخدام طرق الصورة النقطية Bitmap البسيطة،بالطبع)،استخدام البتماب كفرشاة brushأو كائن رسم drawing (كما فعلنا في بعض أمثلة الكود في هذا الفصل سابقاً)،أو الكتابة على نفس سطح صورة نقطية بإرفاق كائن غرافيك Graphicsلها.

إذا كان لديك صورة نقطية في ملف،تستطيع تحميلها بواسطة مشيد فئة البتمابBitmap :

```
Dim niceImage As New Bitmap("c:\LookupItem.bmp")
```

لحفظ كائن الصورة النقطية إلى ملف،استخدم طريقتها Save.

```
niceImage.Save("LookupItem.jpg", Imaging.ImageFormat.Jpeg)
```

مشيد أخر يتيح لك إنشاء صور نقطية جديدة في بتنسيقات متنوعة.

```
'إنشاء صورة نقطية,50-50 باستخدام32بت أومستوى
' لكل من الأحمر والأخضر والأزرق 8 لكل بكسل,
' و8بت بالنسبة لمستوى الشفافية لكل بكسل,من صفر إلى255
        Dim niceImage As New Bitmap(50, 50, Drawing.Imaging.PixelFormat.Format32bppArgb)
```

لرسم صورة على سطح رسومي(غرافيك)،استخدم الطريقة DrawImageلكائن الغرافيك Graphics .

```
e.Graphics.DrawImage(niceImage, leftOffset, topOffset)
```

تلك العبارة ترسم صورة إلى سطح غرافيك كما هو،ولكنه نوع من الملل.تستطيع تمديد الصورة وتحصل على الصورة كما رسمتها،أو حتى إنتاج صورة صغيرة thumbnail.

سأجرب جميع هذه الطرق باستخدام صورة من مشروع المكتبة(الصورة *SplashImage.jpg*)

```
Dim splashImage As New Bitmap("SplashImage.jpg")
' ارسمها بنصف العرض والارتفاع
        e.Graphics.DrawImage(splashImage, New RectangleF(10, 50, splashImage.Width / 2,
splashImage.Height / 2))
' مددها
        e.Graphics.DrawImage(splashImage, New RectangleF(200, 10, splashImage.Width * 1.25,
splashImage.Height / 4))
' ارسم الجزء الأوسط
        e.Graphics.DrawImage(splashImage, 200, 100, New RectangleF(0, splashImage.Height / 3,
splashImage.Width, splashImage.Height / 2), GraphicsUnit.Pixel)
```

يبين الشكل التالي مخرجات المقطع السابق من الكود.ولكن هذا ليس كل الرسم الذي تستطيع عمله.تتضمن الطريقة DrawImage 30 إعادة تعريفoverloads .وذلك سيبقيني مشغول 37 دقيقة على الأقل.



عرض فنك الحقيقي.Exposing Your True Artist

لقد غطينا معظم الميزات الأساسية لـ GDI+المستخدمة لرسم الصور.والآن كل مافي الأمر إصدار أوامر الرسم من أجل الأشكالshapes، الصور images،والنصوص textعلى سطح غرافيكgraphics . معظم الوقت الذي ستمضيه مع الطرق المضمنة في كائن الغرافيك،يوجد القليل من الطرق الهامة،إليك عينات منها:

.طريقة التنظيفClear، تنظف الخلفية بلون معين.

.الطريقة CopyFromScreen.إذا أسقط (غير موجود)مفتاح"تصوير الشاشةPrnt Scrn " على لوحة مفاتيحك،هذه هي الطريقة التي تحتاجها.

.الطريقةDrawArc. رسم جزء من قوس على طول الحافة للقطع الناقص.نقطة الصفر تبدأ عند الساعة الثالثة.قيم المسح للقوس الموجب تتحرك باتجاه عقارب الساعة clockwise، واستخدم قيم المسح السالبة للتحرك بعكس عقارب الساعةcounterclockwise .

. الطرق DrawBezierو DrawBeziers ترسم منحنيات بيزيه، معادلة معتمدة على منحني يستخدم مجموعة من النقاط،زائد مواجهات تقود المنحني خلال النقاط

.الطرق DrawCurve و DrawClosedCurve وFillClosedCurve. ترسم المنحنيات الرئيسيةcardinal (حيث تحدد النقاط مسار المنحني)،مع فرشاة ملئ اختيارية.

.الطرق DrawEllipse وFillEllipse.ترسم قطع ناقص أو دائرة(والتي هي تنوع عن القطع الناقص).

.الطرق DrawIcon، DrawIconUnstretched، DrawImage،DrawImageUnscaled وDrawImageUnscaledAndClipped. طرق مختلفة لرسم صور وأيقونات.

.الطرق DrawLine وDrawLines.ترسم خط أو أكثر مع الكثير من الخيارات لجعل الخطوط جذابةsnazzy .

.الطرق DrawPathوFillPath. سأناقش"مسارات الغرافيك"بعد قليل.

.الطرق DrawPieوFillPie. ترسم حدود شريحة على طول حافة قطع ناقص.

.الطرق DrawPolygon وFillPolygon.ترسم أشكال هندسية نظامية أو غير نظامية بالاعتماد على مجموعة من النقاط.

.الطرق DrawRectangle، DrawRectangles، FillRectangle و FillRectangles.ترسم مربعات squaresومستطيلاتrectangles .

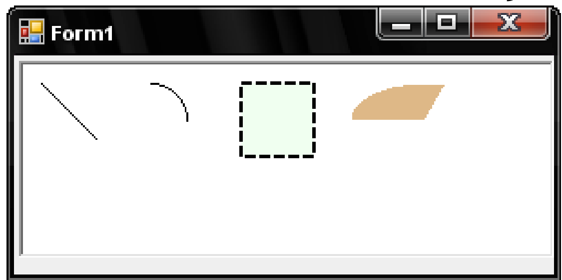
.الطريقة DrawString.نستخدم هذه الطريقة قبل إخراج نص إلى لوحة.

.الطريقة FillRegion.سأناقش القطاعات regions فيما بعد في هذا الفصل.

إليك عينة كود رسم :

```
'خط يمتد من10 ، 10إلى40,40
        e.Graphics.DrawLine(Pens.Black, 10, 10, 40, 40)
'        قوس بـ90درجة باتجاه عقارب الساعة من أجل دائرة بقطر 40 بكسل
        e.Graphics.DrawArc(Pens.Black, 50, 10, 40, 40, 0, -90)
'        مستطيل 40×40 معبئ بخط مقطع
        e.Graphics.FillRectangle(Brushes.Honeydew, 120, 10, 40, 40)
        Using dashedPen As New Pen(Color.Black, 2)
            dashedPen.DashStyle = Drawing2D.DashStyle.Dash
            e.Graphics.DrawRectangle(dashedPen, 120, 10, 40, 40)
        End Using
'        شريحة من مقطع بيضوي
        e.Graphics.FillPie(Brushes.BurlyWood, 180, 10, 80, 40, 180, 120)
```

وهكذا،لقد أخذت فكرة.يبين الشكل التالي مخرجات هذا الكود.



المسارات:الرسومات برؤية برنامج الماكرو. Paths: Drawings on Macro-Vision

تتيح لك فئة GraphicsPathتجميع الكثير من كائنات الرسم البدائية(مثل الخطوط،والأقواس وحتى المستطيلات)في وحدة تجميعية مفردة.هذا المسار الكامل يمكن أن يتم إعادة تشغيله ضمن سطح غرافيك كجامع(شاملmacro).

```
Using thePath As New Drawing2D.GraphicsPath
            thePath.AddEllipse(0, 0, 50, 50)
            thePath.AddArc(10, 30, 30, 10, 10, 160)
            thePath.AddRectangle(New Rectangle(15, 15, 5, 5))
            thePath.AddRectangle(New Rectangle(30, 15, 5, 5))
            e.Graphics.DrawPath(Pens.Black, thePath)
        End Using
```

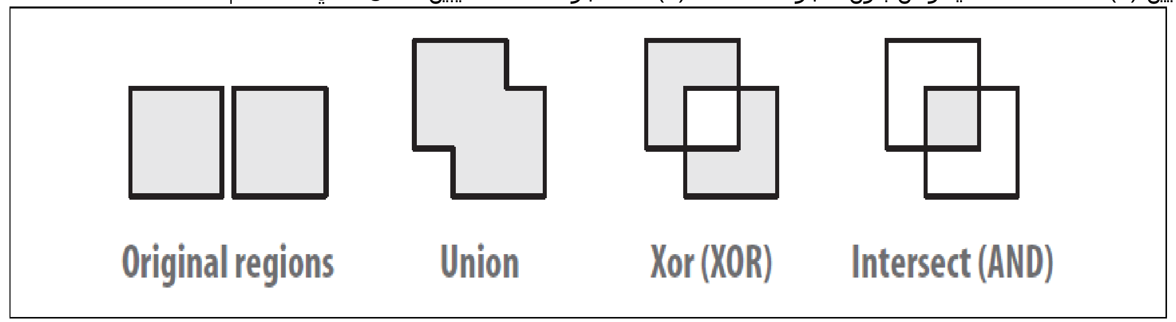


لحسن الحظ توجد استخدامات أخرى لمسارات الغرافيك،بعض منها سأناقشه في المقطع اللاحق.

إبقاءه مقطعي.Keeping It Regional

عادة،عند رسم صورة،فإن لديك رؤية لكامل لوحة الرسم من أجل العمل عليها.(تستطيع رسم الصور والأشكال خارج حدود اللوحة إذا أردت)ولكن في بعض الأوقات تريد أن يظهر فقط جزء من ما ترسم .يستخدم ويندوز هذه الطريقة بنفسه لحفظ الوقت.عندما تحجب نافذة بأخرى،ومن ثم تعرض النافذة المخفية،على التطبيق أن يعيد رسم كل شيء يظهر على النافذة أو الفورم. ولكن لو كان قسم من خلفية تلك النافذة مخفي ومن ثم تم جعلتها مرئية مرةً أخرى،فلما على البرنامج أن يذهب خلال مشاكل رسم كل شيء مرةً أخرى؟في الحقيقة عليه أن يعيد رسم فقط الجزء الذي كان غير مرئي،الجزء الذي كان مخفي في القطاع.

يعين القطاع *region*مساحة من أجل الرسم على سطح ما.والقطاعات غير محددة بالأشكال المستطيلة.تستطيع تصميم قطاع بالاعتماد على أشكال بسيطة.أو تستطيع دمج قطاعات موجودة في قطاعات أكثر تعقيداً.على سبيل المثال،إذا كان لديك قطاعين مستطيلين،تستطيع تشبيكهم وتستحضر قطاع موحد جديد يحتوي على (1) كل القطاعين الأصليين،(2) القطاعات الأصلية ولكن بدون الأجزاء المتداخلة،(3)فقط الأجزاء المتداخلة،يبين الشكل التالي هذا الضم.



خلال عملية الرسم،يتم الإشارة في بعض الأحيان إلى القطاعات كـ"قطاعات قص clipping regions" لأن أي محتوى مرسوم خارج القطاع يتم قصه والتخلص منه.الكود التالي يرسم صورة،ولكن يحجب القطع الناقص في الوسط باستخدام(!tada):مسار غرافيك لتأسيس قطاع قص مخصص:

```
'تحميل الصورة التي سنظهرها أصغر من الصورة الحقيقية
        Dim splashImage As New Bitmap("C:\SplashImage.jpg")
        Dim thePath As New Drawing2D.GraphicsPath()
 '      إنشاءمسار بيضوي والذي يشكل حجم الصورة التي سيتم اخراجها
        thePath.AddEllipse(20, 20, splashImage.Width \ 2, _
        splashImage.Height \ 2)
'          تبديل قطاع القص الأصلي والذي يغطي كامل لوحة الرسم بقطاع مستطيل فقط
        e.Graphics.SetClip(thePath, Drawing2D.CombineMode.Replace)
'          رسم الصورة, والتي سيتم قصها
        e.Graphics.DrawImage(splashImage, 20, 20, _
        splashImage.Width \ 2, splashImage.Height \ 2)
 '        التنظيف
        thePath.Dispose()
```

مخرجات هذا الكود تظهر في الشكل التالي:



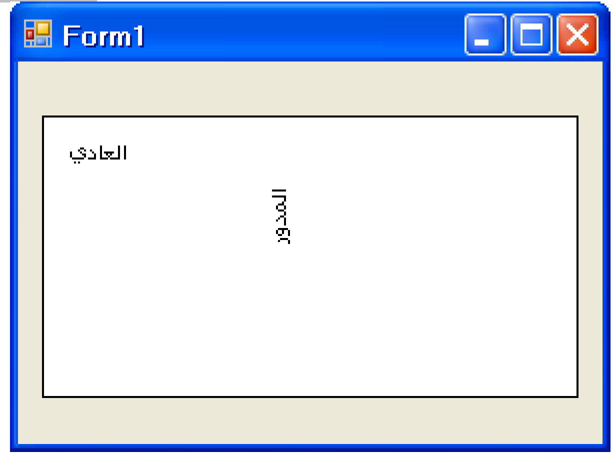
المقاطع مفيدة أيضاً من أجل "اختبار النقر hit testing ".إذا رسمت صورة غير مستطيلة على فورم ،وتريد معرفة متى سينقر المستخدم على الصورة،ولكن ليس على أي بكسل فقط من على الصورة بالكامل،تستطيع استخدام قطاع ما والذي هو نفس شكل الصورة من أجل اختبار نقرات الفارة.

الفتل والقلب بالتحويلات. Twisting and Turning with Transformations

عادةً،أي شيء ترسمه على لوحة الرسوميات(الغرافيك) يتم وضعه مباشرةً على سطح صورة نقطية.فهو يشبه شبكة ضخمة،وأوامر الرسم خاصتك تعمل بشكل أساسي على إسقاط الحبر الملون مباشرةً في كل خلية شبكة.يمنحك كائن الغرافيك القدرة على تمرير أوامر الرسم خاصتك خلال تحويل هندسي قبل أن تظهر مخرجاتها على سطح اللوحة.على سبيل المثال،تحويل الدوران سيعمل على تدوير خطك،أشكالك،ونصوصك بالكمية التي تعينها(بالدرجة)،ومن ثم تطبيق النتيجة إلى سطح.يبين الشكل التالي نتائج الكود التالي،والذي يطبق تحويلين☹1)نقل القطاع(0,0) إلى اليمين 100نقطة(بكسل) وللأسفل 75بكسل،،و(2)إضافة تدوير باتجاه عقارب الساعة من 270درجة.

```
e.Graphics.DrawString("العادي", SystemFonts.DefaultFont, Brushes.Black, 10, 10)
        e.Graphics.TranslateTransform(100, 75)
        e.Graphics.RotateTransform(270)
        e.Graphics.DrawString("المدور", SystemFonts.DefaultFont, Brushes.Black, 10, 10)
```

```
        e.Graphics.ResetTransform()
```



التحويلات تراكمية،إذا طبقت تحويلات متعددة إلى لوحة رسم،فإن أي أوامر رسم ستمر خلال جميع التحويلات قبل الوصول إلى لوحة الرسم.الترتيب الذي تحدث فيه التحويلات هام.إذا تم قلب عبارات TranslateTransform وRotateTransform في الكود الذي شغلته للتو ،فإن التدوير سيغير إحداثيات x, y لعالم لوحة الرسم بالكامل. التحويل اللاحق لـ(100,75)سيتم نقله لأعلى المقطع100بكسل ومن ثم لليمين 75 بكسل.
تتضمن فئة الغرافيك Graphicsهذه الطرق التي تتيح لك تطبيق تحويلات إلى "المنظر العام world view" للوحة الرسم خلال رسم ما يلي:

تحويل التدوير:RotateTransform

يعمل على تدوير المنظر العام(المشهد العام) في بالدرجات في اتجاه عقارب الساعة clockwise degrees من 0 إلى 359.ويمكن للدوران أن يكون موجب أو سالب.

تحويل متدرج: ScaleTransform

يضع عامل تدريج(موازنة) scaling factor لجميع الرسوم.بشكل أساسي،يعمل على زيادة أو إنقاص حجم شبكة لوحة الرسم عند الرسم.تغير الموازن يؤثر على عرض الأقلام.إذا وازنة الكائنات بالعامل 2،لا تظهر فقط الفراغات بحيث تصبح مرتين أبعد عن بعضها،ولكن جميع الأقلام ترسم أسمك بمرتين أيضاً عن الغير موازنة.

تحويل النقل.TranslateTransform

يعيد وضع الأعضاء بالاعتماد على تعويضات x وy.

التحويل المتعدد.MultiplyTransform

نوع من طرق التحويل الرئيسية master transformationالتي تتيح لك تطبيق تحويل من خلال كائن قالب.إن له خيارات أكثر من التحويلات القياسية المضمنة في كائن الغرافيك.على سبيل المثال، تستطيع تطبيق تحويل قص والذي يحرف جميع المخرجات في مستطيل إلى تغير من نوع متوازي مستطيلات-rectangle-to parallelogram.

إعادة وضع التحويل ResetTransform

يعمل على إزالة جميع التحويلات المطبقة من لوحة رسم ما.

حفظ .Save

يحفظ الحالة الحالية لسطح الرسوميات المحولة(أو الغير محولة) إلى كائن من أجل الاسترداد اللاحق.وهذا يتيح لك تطبيق بعض التحويلات،حفظها،تطبيق تحويلات أكثر، ومن ثم استرداد المجموعة المحفوظة،إزالة أي تحويلات مطبقة منذ المجموعة التي تم حفظها.

الاسترداد.Restore

استرداد مجموعة محفوظة من التحويلات.

تحسين الأدوات من خلال حامل الرسم.Enhancing Controls Through Owner Draw

يتم تضمين الكثير من ميزات رسم الـ GDI+ في الدوت نت،ولكن نراه هنا قد يكون كافي لرواية عطشك whet your appetite.تستطيع عمل الكثير من الرسم الخيالي بالـGDI+، ولكن دعنا نواجهها:أنت وأنا مبرمجين،ولسنا فنانين ، ولكن لحسن الحظ،توجد أشياء شبه فنية تطبيقية تستطيع عملها بالـGDI+. وواحدة من الميزات المهمة هي حامل الرسم(المالكowner draw)،مشاركة مسؤوليات الرسم بين الأداة وأنت،المبرمج(أنت "المالك" owner")،تدعم أداة الصندوق المركب ComboBox مالك الرسم owner drawing للبنود المستقلة في القسم المنسدل drop-down portion للقائمة. لنعمل على إنشاء أداة صندوق مركب،والتي تعرض أسماء الألوان،متضمنةً عينة صغيرة من الألوان إلى يسار الاسم.اعمل على إنشاء تطبيق ويندوز جديد،وأضف أداة صندوق مركب سمها ComboBox1 إلى النموذج Form1.
اعمل التغيرات التالية لأداة الصندوق المركبComboBox1 :

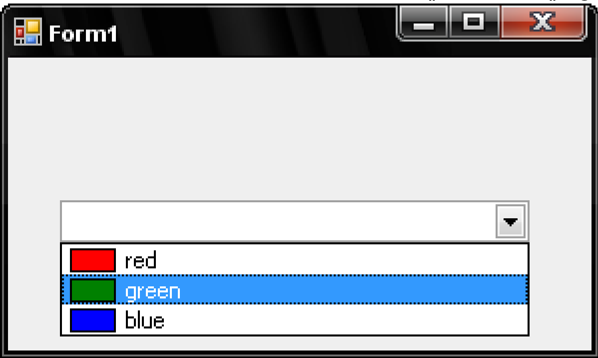
1. غير خاصية DropDownStyleإلى DropDownList.
2. غير خاصية DrawModeإلى OwnerDrawFixed.
3.بدل خاصية Items،بإضافة أسماء ألوان متعددة كنصوص متميزة في نافذة محرر تجميع النصوص String Collection. إني عملت على إضافة الأحمر،الأخضر،والأزرق.

والآن أضف الكود التالي إلى منطقة الكود المصدري لفئة الفورم Form1.

```
  Private Sub ComboBox1_DrawItem(ByVal sender As Object, ByVal e As System.Windows.Forms.DrawItemEventArgs)
Handles ComboBox1.DrawItem
' تجاهل حالة عدم الاختيار
        If (e.Index = -1) Then Return
' إنشاء فرشاة لعرض اللون,بالاعتماد على اسم البند
```

```
        Dim colorBrush As New SolidBrush(Color.FromName(CStr(ComboBox1.Items(e.Index))))
اختيارة مدع أو البند هذا إما اختيار على تعتمد اللون نتشكيلة فرشاة إنشاء       '
        Dim textBrush As Brush
        If ((e.State And DrawItemState.Selected) = DrawItemState.Selected) Or ((e.State And
DrawItemState.HotLight) = _
        DrawItemState.HotLight) Then
            textBrush = New SolidBrush(SystemColors.HighlightText)
        Else
            textBrush = New SolidBrush(SystemColors.ControlText)
        End If
اللون عرض لمنطقة الشكل على الحصول        '
        Dim colorBox As New Rectangle(e.Bounds.Left + 4, e.Bounds.Top + 2, (e.Bounds.Height - 4) * 2,
e.Bounds.Height - 4)
.مختارة الغير أو المختارة الخلفية رسم        '
        e.DrawBackground()
.مخصص لون منطقة رسم        '
        e.Graphics.FillRectangle(colorBrush, colorBox)
        e.Graphics.DrawRectangle(Pens.Black, colorBox)
اللون يمين إلى اللون اسم رسم        '
        e.Graphics.DrawString(CStr(ComboBox1.Items(e.Index)), ComboBox1.Font, textBrush, 8 + colorBox.Width,
e.Bounds.Top + ((e.Bounds.Height - ComboBox1.Font.Height) / 2))
.الحاجة عند,البند حول المختار المستطيل رسم        '
        e.DrawFocusRectangle()
.التنظيف        '
        textBrush.Dispose()
        colorBrush.Dispose()
    End Sub
```

شغل التطبيق ولاحظ الصندوق المركب،كما هو مبين في الشكل التالي:



أساسيات تقديم النوافذ.Windows Presentation Foundation

أداة ميكروسوفت الأخيرة لبناء،واجهات مستخدم قريبة الإنتاج وفعالة هي أساسيات تقديم النوافذWindows Presentation Foundation، أو WPF.وكما في لينكو، WPF تدمج meldsعدة تقنيات مختلفة مع بعضها ضمن كل موحدunified whole . بعض من هذه التقنيات كانت معنا منذ عدة سنين،مثل نظام ميكروسوفت ثلاثي الأبعاد Microsoft's Direct3D system والذي يعرض ويعالج العناصر ثلاثية الأبعاد3D. تختصر WPFجميع هذه التقنيات،وتجعلها متاحة من خلال اللغة التصويرية المعتمدة على XML-based descriptive language XML والمعروفة بـ XAML(لغة ترميز التطبيق القابلة للتوسع eXtensible Application Markup Language).

تتضمن WPFميزات وعناصر تتعامل مع عدة مناطق تقديم،من ضمنها الأدوات التي على الشاشةon-screen controls، الرسوميات ثنائية البعد2D drawings (مثل GDI+)،الرسوميات (الغرافيك)ثلاثي البعد3D graphics (من الأبعاد الثلاثية الاتجاه Direct3D)،الصور الثابتةstatic images (مثل صور JPEG)،الوسائط المتعددة التفاعلية interactive multimedia (الفيديو و الأوديو)،وتقديم مستند "ماتراه هو ما تحصل عليه WYSIWYG "(مشابهة لمستندات PDF).يمكن جعل العناصر المستقلة وكامل واجهة المستخدم حيوية animatedبشكل آلي،أو في حالة الاستجابة لتفاعل المستخدم.

عندما يحين الوقت لعرض محتوىWPF، تستطيع إحضاره إلى المستخدم في عدة طرق مشتركة. يمكن أن يتم بناء ملفات XAMLوما يتعلق بها من كود دوت نت في تطبيق قائم بذاته.وهو مشابه كثيراً لتطبيق نماذج ويندوز للدوت نت النموذجي،ولكن مع العديد من الهبات المدهشة للمظهر والتي في التطبيق العادي أو القياسي لايمكن للمطورين الوصول إليها.

يمكن أيضاً استخدام WPFلإنتاج تطبيقات معتمدة على الويب يتم استضافتها على متصفح الانترنتbrowser. في الحقيقة،العناصر التي تصممها للاستخدام في تطبيقات تخطيط سطح المكتب desktop-style applications يمكن أن يتم استخدامها على الويب بشكل عام بدون أي تعديل.وكما تتوقع،قيود الأمن التي تضع المشطات على بعض الأشياء التي تستطيع عملها عند التشغيل في مستضيف من هذا النوع.

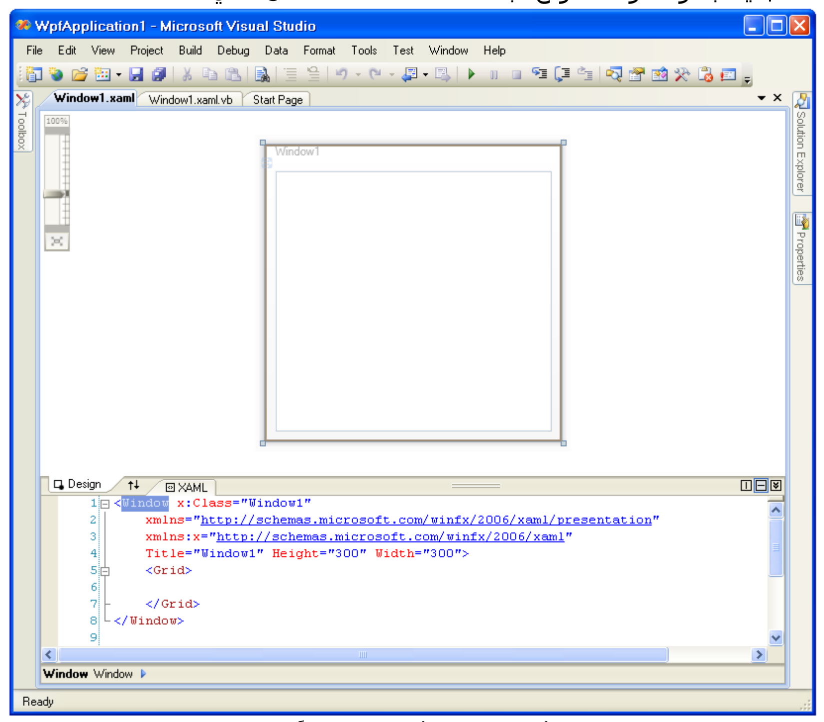
برامج WPFالمعتمدة على المتصفح تتطلب أن يتم تثبيت كل من مكتبات إطار عمل الدوت نت و WPF مع إمكانية الوصول إليها على شبكة(أو كمبيوتر)العميل.تبني ميكروسوفت الكثير من تلك التقنيات وتحزمها في منتج يدعى *Silverlight*.المصمم للكمبيوتر والذي هو كثير الشبه بمنصة فلاش أدوبي Adobe's Flash،ونظام جافا إف إكس JavaFX system لشركة سان ميكروسيستمSun Microsystem،سيسمح Silverlightفي النهاية للمحتوى المرتبط لكل من XAMLو دوت نت لأن يعمل على منصات بدون نوافذnon-Windows platforms، مثل حاسوب ماكينتوشMacintosh .

تشكيلة ثالثة تستخدم مجموعة جزئية من XAMLلتعريف مستند ثابت شبيه بـ بي دي إفPDF-like static document. مثل المستندات التي تعرف بـمستندات *XPS*(توصيف الورقي(أو المستندي) XML Paper Specification)،وعملياً هي ملفات ZIPتحوي على جميع صفحات XAML،الرسومياتgraphics، وعناصر الصفحة الأخرى في ملفات متميزة ضمن الأرشيف.

WPF و XAML.

واحدة من العلامات المميزة في تصميم التطبيق ضمن WPFهو الفصل بين المنطق والتقديم.وهذا الهدف مشترك بين العديد من التقنيات الجديدة،من ضمنها XMLو ASP.NET. كل منطق التطبيق- أي جميع الأحداث التي يتم إطلاقها من قبل مدخلات المستخدم وأفعال النظام- يتم كتابتها في كود دوت نت قياسي.ويمكن أن تظهر أيضاً واجهة المستخدم الغرافيكية كـــكود دوت نت،مع الكائنات المنشئة خارج WPFالمخصص لفضاء الأسماءSystem.Windows. ولكن الأكثر شيوعاً تصميم عناصر واجهة المستخدم والأدوات المشتقة من خلال XAML،وتخطيط XMLالذي يمكن إنتاجه من قبلك في الفيجوال أستوديو أو المفكرة،أو بواسطة أدوات ثانوية. لأن محتوى XAMLيمكن أن يتم بناءه خارج تطبيقه،فتخصص تصميم واجهة المستخدم مع معرفة برمجية محدودة يمكن أن تبني مكونات واجهة المستخدم بشكل منفصل عن عمل المطور على منطق التطبيق.توفر ميكروسوفت أداة لمثل هذه المصممات،تدعى *Microsoft Expression Blend*،التصميم التعبيري Expression design واحد من المنتجات القليلة.توفر بعض الشركات الأخرى أدوات إنتاج محتوى XAMLغني.

تتيح لك الفيجوال أستوديو إنشاء تطبيقات WPFكاملة بالاعتماد على محتوى XAML.لبناء تطبيقWPF ، ابدأ مشروع فيجوال أستوديو جديد،أنشئ مشروع جديد new project،واختر من صندوق حوار مشروع جديد New Project قالب تطبيق WPF Application ضمن نوع مشروع الفيجوال بيسك.ومن ثم انقر موافقOK. تعمل ويندوز على إنشاء مشروع نماذج WPFجديد مباشرةً،عارضةً نموذج البدء، Window1 شاهد الشكل التالي.



واجهة المستخدم معرفة بالكامل بواسطة مجموعة XMLالمعروضة أسفل النموذج(أو النافذة).حالياً،تعرف نافذة التطبيق النهائية نفسها.

```
<Window x:Class="Window1"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Window1" Height="300" Width="300">
    <Grid>
    </Grid>
</Window
```

يعرف هذا الكود النافذة، Window1،والتي لديها فئة System.Windows.Window حقيقية.مواصفات وسم نافذةXAML، تتضمن العنوانTitle، الارتفاع Height والعرضWidth، مرتبطة بالخاصيات propertiesلنفس الاسم في فئة النافذةWindow. دعنا نضيف بعض الإثارة لهذه الفورم،سأعمل على إضافة زر يتضمن قوس قزح rainbowعلى واجهته،زائد إضافة لمعة صفراء حول الزر،سأعمل أيضاً على إضافة معالج الحدث الذي يظهر صندوق رسالة،كما مع تطبيق نماذج ويندوز القياسي،فإنك تستخدم الأدوات في صندوق الأدوات لبناء الفورم.سأعمل على سحب زر إلى سطح الفورم واستخدم منطقة نص XAMLلجلب حياة جديدة إلى النافذة.

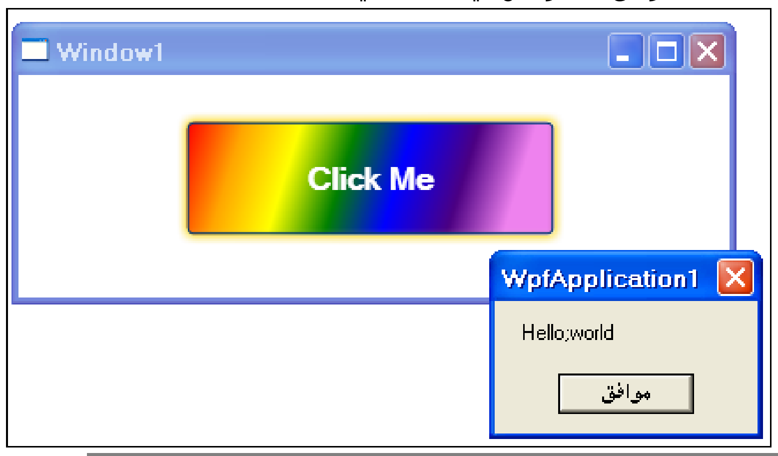
```
<Window x:Class="Window1"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Window1" Height="160" Width="411">
    <Grid>
        <Button Margin="95,26,99,35" Name="Button1">
            <Button.Foreground>White</Button.Foreground>
            <Button.FontSize>18</Button.FontSize>
            <Button.FontWeight>Bold</Button.FontWeight >
            <Button.Background>
                <LinearGradientBrush>
                    <LinearGradientBrush.GradientStops>
                        <GradientStopCollection>
                            <GradientStop Color="Red" Offset="0" />
```

```
                        <GradientStop Color="Orange" Offset="0.1425"/>
                        <GradientStop Color="Yellow" Offset="0.285"/>
                        <GradientStop Color="Green" Offset="0.4275"/>
                        <GradientStop Color="Blue" Offset="0.57"/>
                        <GradientStop Color="Indigo" Offset="0.7325"/>
                        <GradientStop Color="Violet" Offset="0.875"/>
                    </GradientStopCollection>
                </LinearGradientBrush.GradientStops>
            </LinearGradientBrush>
        </Button.Background>
        <Button.BitmapEffect>
            <OuterGlowBitmapEffect />
        </Button.BitmapEffect> Click Me
    </Button>
  </Grid>
</Window>
```

سأعمل أيضاً على إضافة معالج حدث،باستخدام نفس الطريقة المستخدمة في مشاريع نماذج ويندوز وذلك بالنقر المزدوج على الزر ليظهر معالج حدث نقر الزر ولنكتب بداخله صندوق رسالة كما يلي:

```
Class Window1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.Windows.RoutedEventArgs) Handles Button1.Click
        MsgBox("Hello;world")
    End Sub
End Class
```

شغل هذا البرنامج بالضغط على المفتاح F5يعمل البرنامج كما هو مبين في الشكل التالي:



تحسين ا لفئات بالمواصفات.Enhancing Classes with Attributes

مواصفات تعديل فئة Class-modifying attributes areشيء ما ناقشناه في الفصل الأول،وليس لها علاقة بالـ GDI+.أردت فقط أن أذكرك بها بما أننا سنستخدمها في كود مشروع هذا الفصل.

تظهر مواصفات تعديل فئة أو عضو قبل تعريف الفئة أو العضو ،وضمن أقواس حادة.يعمل الكود التالي على إلحاق المواصفة ObsoleteAttribute إلى الفئة SomeOldClass.

```
<ObsoleteAttribute()> Class SomeOldClass
    ' ضع تفاصيل تتعلق بالفئة هنا
End Class
```

بإمكانك أن لاتضع الجزء" Attribute "للمواصفة إذا كان اسم المواصفة لا يتعارض مع كلمة محجوزة للفيجوال بيسك (أي أن تكتب اسم المواصفة فقط Obsolete).

تظهر المواصفات كميتاداتا(توصيف بيانات metadata)في المجمع النهائي المترجم،ويتم استخدامها بواسطة الفئات والتطبيقات والتي ،بشكل تصميمي،تستخرج مدلول(مغزى) مواصفات معينة.في كود هذا الفصل،سنستفيد من الأداةPropertyGrid، الأداة التي تنفذ خاصيات لوحة ضمن بيئة تطوير الفيجوال أستوديو،وعلى الأغلب تستخدم لتعديل خاصيات أداة وفورم،واستخدام هذه الأداة متاح لك في تطبيقاتك الخاصة.لاستخدامها،اسند حالة أو نسخة فئة إلى خاصية الأداة SelectedObject،ومن ثم،بشكل سحري،تظهر جميع خاصيات الكائن في قائمة خاصيات الأداة..

ولكنها ليست مفضلة دائماً،يمكن أن يكون لكائنك خاصيات يجب أن لا تظهر أو يجب عدم عرضها.لقد تم تصميم الأداة كي تكون شاملة أو عامة،ولا تعرف شيء حول احتياجات كائنك،لذلك فهي لا تعرف أي خاصيات يجب عدم تضمينها.هذا كل شيء فهي لاتعرف حتى تخبرها أنت من خلال المواصفات.بإضافة مواصفات معينة لخاصيات فئتك،فإنك تخبر الأداة PropertyGridكيف تعامل أعضاء كائنك.على سبيل المثال، تخبر المواصفة BrowsableAttributeالأداة PropertyGridلتضمين(في حال كانت True)أو إخراج(في حال كانت False) خاصية.

```
<Browsable(False)> Public Property SecretProperty( ) As String...
```

سأقدم لك تفاصيل إضافية حولها عندما نستخدم الأداة PropertyGridفيما بعد في هذا الفصل.

مشروع.Project

لقد استخدم مشروع المكتبة ميزات الـ GDI+منذ اللحظة الأولى عند إنشاء أول فورم في المشروع.ولكن كل هذا كان بدون تدخل منك بما أنه مضمن في إطار العمل.والآن حان وقت وضع لمساتك الخاصة،كمبرمج،من أجل إضافة الـ GDI+الخاصة للتطبيق.في هذا كود مشروع هذا الفصل،سنعمل على استخدام GDI+ لتحسين العرض العادي لأداة من خلال حامل ميزات الـرسم.زائد،أننا سنعمل على البدء أخيراً بتنفيذ بعض ميزات كود التعريف bar code التي أغريتك بها في فصول سابق.

Install the Bar Code Font.تثبيت خط كود التعريف

إذا لم تحصل حتى الآن على خط كود التعريف bar code font،فهذا الوقت المناسب لعمله.الميزات المضمنة في كود مشروع هذا الفصل سيطلب منك استخدام خط، يمكنك شراء خط كود تعريف احترافي.ولكن تأكد من أن الخط الذي حصلت عليه هو نوع خط حقيقي.(راجع الملحق A)

Using Owner Draw.استخدام حامل الرسم

في الفصل السابق،عملنا على إضافة الفورم *ItemLookup.vb* مع عروضها المتعددة لبنود المكتبة.واحد من هذه العروض المضمنة الأداةMatchingItems ، صندوق قائمة listboxمتعددة الأعمدة تعرضAuthor/Name ، استدعاء عددNumber، وأعمدة لنوع الميديا(أو الوسيطةMedia Type).على الرغم من أننا خزّنا البيانات الخاصة بعمود ضمن كل بند سابق،لم نعرض حقاً الأعمدة المستقلة للمستخدم.

الشيء الذي يخص عرض القوائم المتعددة الأعمدة و النصوص الأخرى المحددة بفراغات هو أن بعض النص يكون محصور بمنطقته رسمية مسبقاً إذا سمحت له بذلك. على سبيل المثال،يمكن للنص في عمود قائمة واحد أن يتداخل إلى نص عمود أخر.في مثل هذه الحالات،فقد أصبح من العرف بتر المحتوى الإضافي واستبداله بعلامة الإضمار ellipsis ("...").لذلك فإننا بحاجة إلى روتين سيحدد فيما إذا كان النص طويل جداً بالنسبة لمنطقة عرضه،وعمل بتر وإضمار عند الحاجة.أضف الطريقة FitTextToWidth إلى كود الوحدة البرمجية*General.vb* .

```
Public Function FitTextToWidth(ByVal origText As String, ByVal pixelWidth As Integer, ByVal canvas
As System.Drawing.Graphics, _
            ByVal useFont As System.Drawing.Font) As String
'        تقديم نص سلسلة حرفية,التأكد من أنها تتناسب في اتساع البيكسل المخصص.
'        بتر وإضافة علامات الإضمار عند الحاجة.
        Dim newText As String
        newText = origText
        If (canvas.MeasureString(newText, useFont).Width() > pixelWidth) Then
            Do While (canvas.MeasureString(newText & "...", useFont).Width() _
                    > pixelWidth)
                newText = Left(newText, newText.Length - 1)
                If (newText = "") Then Exit Do
            Loop
            If (newText <> "") Then newText &= "..."
        End If
        Return newText
    End Function
```

إن النموذج *ItemLookup* لديه زر "رجوع" الشبيه بمتصفح الانترنت مع قائمة سياق(أو انسدال) للمدخلات الحديثة.البنود المضافة إلى هذه القائمة يمكن أن تتضمن عنوان كتاب وأسماء مؤلفين طويل جداً.دعنا نستخدم الطريقة FitTextToWidthلتحديد حجم بنود النص في هذه القائمة.افتح الكود المصدري للنموذج ItemLookup واعمل على إيجاد الطريقةRefreshBackButtons. واستبدل السطر التالي من الكود .

```
whichMenu.Text = scanHistory.HistoryDisplay
```

بالسطر التالي.

```
whichMenu.Text = FitTextToWidth(scanHistory.HistoryDisplay, Me.Width \ 2, useCanvas,whichMenu.Font)
```

هذا سيعمل على تقيد نص بند قائمة إلى نصف اتساع الفورم،وهذا يبدو معقولاً بالنسبة لي.ولكن المتغير `useCanvas` جديد،لذلك أضف التصريح عن هذا المتغير في أعلى الطريقة RefreshBackButtons.

```
Dim useCanvas As Drawing.Graphics = Me.CreateGraphics()
```

ونحتاج أيضاً إلى التخلص من لوحة الرسم الغرافيكية canvas التي عملنا على إنشاءها من خلال التصريح عن المتغير في السطر السابق، ويتم ذلك عند نهاية الطريقة، لذلك أضف الكود التالي قبل سطر نهاية الطريقة `End Sub` .

```
useCanvas.Dispose()
```

والآن لنعالج حامل رسم owner draw بنود قائمة.تسمح لك أدوات صندوق القائمة ListBoxاستخدام كود رسم خاص من أجل كل بند مرئي في القائمة.لديك خيارين عندما تدير البنود المرسومة بنفسك:تستطيع الحفاظ على كل بند بارتفاع ثابت،أو تستطيع جعل كل بند قائمة بارتفاع مختلف بالاعتماد على محتوى ذلك البند.في أداة صندوق القائمة MatchingItems،سنعمل على استخدام نفس الارتفاع من أجل كل بنود قائمة.لتمكين نمط حامل الرسمowner draw، افتح محرر تصميم الفورم ItemLookup،اختر أداة صندوق القائمة MatchingItemsعلى الفورم أو من خلال لوحة الخاصيات،وغير خاصيتها DrawModeإلى OwnerDrawFixed.

كل بند قائمة مطابق سيتضمن سطرين من البيانات☹1)عنوان البند المطابق،بخط غامق،و(2)الأعمدة الثلاث للمؤلف،رقم الاستدعاء،نوع الميديا(أو الوسيطة)البيانات. أضف الكود التالي إلى معالج حدث تحميل الفورم الذي يحدد كامل ارتفاع كل بند قائمة،وموضع السطر الثاني ضمن كل بند.

```
Private Sub ItemLookup_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
'        تحضير الفورم
        Dim formGraphics As System.Drawing.Graphics
'        وضع ارتفاع البنود الافتراضي في صندوق القائمة
        ' matching items
        formGraphics = Me.CreateGraphics()
        MatchingItems.ItemHeight = CInt(formGraphics.MeasureString("A" & vbCrLf & "g",
MatchingItems.Font).Height()) + 3
        SecondItemRow = CInt(formGraphics.MeasureString("Ag", _
            MatchingItems.Font).Height()) + 1
```

```
        formGraphics.Dispose()
    End Sub
```

استخدمت النص " Ag "لأتأكد من أن الارتفاع يتضمن جميع الخطوط المرتفعة ascenders والنازلة descendersعن مستوى السطر(الأجزاء التي تبرز للأعلى أو للأسفل من معظم الحروف). أعتقد أن الحساب سيتضمن تلك القيم حتى ولو استخدمت " mm " من أجل النص،ولكن من الأفضل،وضع الخاصية MatchingItems.ItemHeightلتشير إلى حجم جميع البنود في القائمة.إذا قررت استخدام بنود متغيرة الارتفاع بدلا عن بنود ذات ارتفاع ثابت،كان يجب علينا معالجة حدث MeasureItem.من خلال البنود ذات الارتفاع الثابت نستطيع تجاهل ذلك الحدث،والانتقال إلى الحدث الذي يعمل رسم حقيقي: DrawItem.

إليك ما سيعمله الكود من أجل كل بند قائمة(1☹إنشاء كائنات الفراشي والخط الذين سنستخدمها في الرسم،(2)رسم السلسلة الحرفية للنص على لوحة رسم بند قائمة،و(3) التنظيف أو الإزالة.بما أنه يمكن أيضاً أن يتم اختيار بنود قائمة أو عدم اختيارها،بالتالي سنستدعي بعض الطرق الموفرة من قبل إطار العمل لرسم عناصر الواجهة والخلفية المناسبة والتي تشير إلى اختيارات بند .

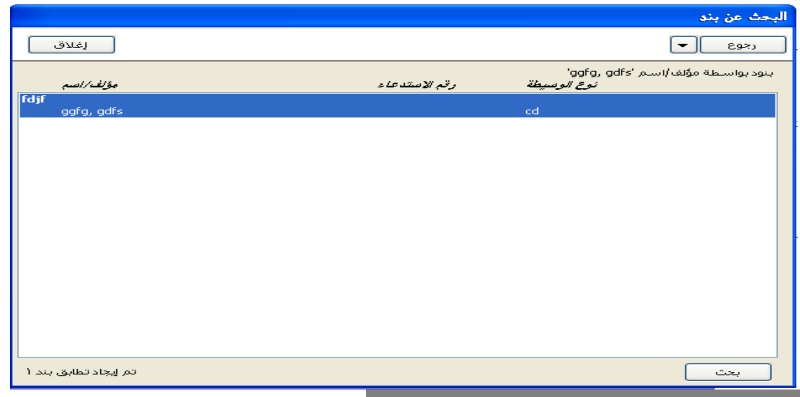
عندما نرسم عدة أعمدة نص ، من المحتمل أن يكون واحد من الأعمدة طويل جداً ويتداخل مع منطقة العمود التالي.وكان هذا هو السبب في كتابة الدالة FitTextToWidth سابقاً. ولكن بدوره يتضمن الـ GDI+سابقاً ميزة تعمل على إضافة علامة الإضمار للنص عند الموضع اليميني عندما لا يكون مناسباً. وهذه الميزة موجودة في فئة تدعى StringFormatفي خاصيتها Trimming،اسند هذه الخاصية إلى EllipsisCharacter واستخدمها عند رسم نص سيتم بتره ليصبح مناسب في الاتساع.عندما نرسم النص على لوحة الرسم،سنعمل على توفير مستطيل rectangle يخبر النص عن حدوده.إليك الكود الأساسي المستخدم لرسم عمود واحد من النص المبتور.

```
Dim ellipsesText As New StringFormat
ellipsesText.Trimming = StringTrimming.EllipsisCharacter
e.Graphics.DrawString("نص ما طويل", e.Font, someBrush, New Rectangle(Left, Top, Width, Height),
ellipsesText)
```

الكود الذي سنستخدمه لرسم كل بند قائمة في القائمة MatchingItemsسيستخدم كود مشابه لهذا.لذلك لنعمل على إضافة هذا الكود الآن إلى معالج حدث MatchingItems.DrawItem.

```
Private Sub MatchingItems_DrawItem(ByVal sender As Object, ByVal e As
System.Windows.Forms.DrawItemEventArgs) Handles MatchingItems.DrawItem
' رسم البنود المتطابقة على سطرين
        Dim itemToDraw As MatchingItemData
        Dim useBrush As System.Drawing.Brush
        Dim boldFont As System.Drawing.Font
        Dim ellipsesText As StringFormat
' رسم خلفية البند
        If (CBool(CInt(e.State) And CInt(DrawItemState.Selected))) Then useBrush =
SystemBrushes.HighlightText Else useBrush = SystemBrushes.WindowText
        e.DrawBackground()
' سيستخدم العنوان خط النسخة الغامقة من الخط الرئيسي
        boldFont = New System.Drawing.Font(e.Font, FontStyle.Bold)
' الحصول على البند لرسمه
        itemToDraw = CType(MatchingItems.Items(e.Index), MatchingItemData)
        ellipsesText = New StringFormat
        ellipsesText.Trimming = StringTrimming.EllipsisCharacter
' رسم نص بند
        e.Graphics.DrawString(itemToDraw.Title, boldFont, useBrush, _
            New Rectangle(0, e.Bounds.Top, ItemColEnd.Left - MatchingItems.Left, _
            boldFont.Height), ellipsesText)
        e.Graphics.DrawString(itemToDraw.Author, e.Font, useBrush, _
            New Rectangle(ItemColAuthor.Left, e.Bounds.Top + SecondItemRow, _
            ItemColCall.Left - ItemColAuthor.Left - 8, e.Font.Height), ellipsesText)
        e.Graphics.DrawString(itemToDraw.CallNumber, e.Font, useBrush, _
            New Rectangle(ItemColCall.Left, e.Bounds.Top + SecondItemRow, _
            ItemColEnd.Left - ItemColType.Left, e.Font.Height), ellipsesText)
        e.Graphics.DrawString(itemToDraw.MediaType, e.Font, useBrush, _
            New Rectangle(ItemColType.Left, e.Bounds.Top + SecondItemRow, _
            ItemColType.Left - ItemColCall.Left - 8, e.Font.Height), ellipsesText)
' إذا كان التركيز على صندوق القائمة, ارسم مستطيل التركيز
        e.DrawFocusRectangle()
        boldFont.Dispose()
    End Sub
```

كما ترى من السهل رسم أي شيء تريد في بند صندوق قائمة، في هذا الكود،المخرجات الفعلية للوحة الرسم بواسطة الـ GDI+المكافئ لعبارات DrawString الأربع.على الرغم من أن قاعدة بيانات المكتبة لا تدعم هذا،كان بإمكاننا تضمين صورة عن كل بند في قاعدة البيانات،وعرضه في صندوق القائمة هذا،إلى يسار العنوان تماماً،وأيضاً،تسمح الاستدعاءات e.DrawBackground و e.DrawFocusRectangle للأداة من التعامل مع البند الصحيح المعلم(أو المظلل)(على الرغم من أنني لم اختار فرشاة نص مناسبة).يبين الشكل التالي نتائج عملنا الجاد.



تصميم البار كود(كود التعريف).Bar Code Design

يتضمن مشروع المكتبة دعم شامل للصاقات كود التعريف.لقد زرت العديد من المكتبات وقارنت أكواد التعريف المضافة لكل من بنود المكتبة(مثل الكتب) وكروت معرف الزبون.وجدت أن التشكيلة كانت واسعة بحيث يضيق حيز حل محدد مسبق أن يتسع لها.وهكذا،يسمح تطبيق المكتبة للمدراء أو أمناء المكتبة من تصميم صفيحة لصاقات كود تعريف تلبي حاجاتهم الخاصة.(هناك أعمال لبيع لصاقات كود تعريف وكروت مطبوعة مسبقاً لمكتبات لاتطبع هذه الكروت واللصقات يدعم التطبيق أيضاً هذه الطريقة بما أن إنتاج البار كود وإسناد البار كود إلى البنود خطوتين منفصلتين).لدعم تصميم بار كود شامل،سنعمل على إضافة مجموعة من الفئات المصممة ونموذجين إلى التطبيق):

BarcodeItemClass.vb

يحتوي ملف هذه الفئة ست فئات متميزة،واحد منها هو الفئة القاعدية من أجل الفئات الخمس الباقية.تصمم الفئات المشتقة عناصر النص الثابتة،صور البار كود،أرقام البار كود،الأسطر،والمستطيلات التي سيضيفها المستخدم إلى سطح لصاقة بار كود واحد..

BarcodePage.vb

هذا النموذج هو فورم محرر مشتق من BaseCodeForm،نفس الفورم القاعدي المستخدم من أجل محررات الكود المتنوعة في التطبيق.تعين هذه الفورم ترتيب صفيحات اللصاقة.من المحتمل أن يشتري المستخدم صفيحات لصاقة من مخزن مكتب محلي.بإدخال رقم صفوف وأعمدة اللصاقة،حجم كل لصاقة ،وأي فراغ بين وحول كل لصاقة،يمكن للمستخدم أن يصمم إلى حدٍ بعيد أي صفيحات لصاقة نظامية.

BarcodeLabel.vb

محرر أخر معتمد على BaseCodeForm،تتيح هذه الفورم للمستخدم من تصميم لصاقة بار كود مفرد بإضافة نص،البار كودات،الأسطر،والمستطيلات إلى منطقة العرض.في فصل لاحق، سنعمل على إضافة طباعة اللصاقة،حيث يتم دمج اللصاقات والصفحات مع بعضها في عمل طباعة رائع.

بما أن هذه الملفات الثلاث تتضمن حوالي 2,000سطر من الكود المصدري،سأبين لك فقط مفاتيح المقاطع لكل واحد منها.فقد عملت على إضافة هذه الملفات الثلاث إلى كود المشروع، لذلك لنبدأ بالملف*BarcodeItemClass.vb*. فهو يعرف كل نوع لبند معروض سيعمل المستخدم على إضافته إلى قالب اللصاقة في الفورم .إليك كود ملخص للفئة القاعدية BarcodeItemGeneric.

```
Imports System.ComponentModel
Public MustInherit Class BarcodeItemGeneric
    <Browsable(False)> Public MustOverride ReadOnly Property ItemType() As String
    Public MustOverride Overrides Function ToString() As String
End Class
```

لايحدث الكثير هنا،تعرف الفئة عضوين مطلوبين: خاصية نصية للقراءة فقط مسماة ItemType،ومتطلبات توفرها الفئات المشتقة من أجل تنفيذ خاص بها الطريقة ToString. الفئات الخمس الأخرى المشتقة في هذا الملف تحسن الفئة القاعدية لدعم أنواع مميزة لعرض العناصر المضمنة على لصاقة بار كود.لنلقي نظرة سريعة على واحدة من هذه الفئات،الفئة BarcodeItemRect،التي تسمح بتعبئة مستطيل بشكل اختياري ليظهر على لصاقة الكود،وتتضمن أعضاء خاصة تقتنص تفاصيل المستطيل.

```
Public Class BarcodeItemRect
' تضمين عنصر المستطيل الأساسي في لصاقة البار كود
    Inherits BarcodeItemGeneric
' تخزين خاص للمواصفات
    Private StoredRectLeft As Single
    Private StoredRectTop As Single
    Private StoredRectWidth As Single
    Private StoredRectHeight As Single
    Private StoredRectColor As Drawing.Color
    Private StoredFillColor As Drawing.Color
    Private StoredRectAngle As Short
```

يتضمن باقي الفئة خاصيات توفر واجهة عامة لهذه الأعضاء الخاصة.إليك الكود من أجل الخاصية FillColor:

```
<Browsable(True), DescriptionAttribute("وضع لون تعبئة المستطيل")> _
    Public Property FillColor() As Drawing.Color
' لون التعبئة
        Get
            Return StoredFillColor
        End Get
        Set(ByVal Value As Drawing.Color)
            StoredFillColor = Value
        End Set
```

```
    End Property
```

مثل معظم الخاصيات الأخرى،فهي تضع وتستخرج القيمة الخاصة المناسبة.يتضمن تصريحها مواصفتين سيتم قراءتهما بواسطة الأداة PropertyGrid فيما بعد.تقول الخاصية Browsable"نعم،ضمن هذه الخاصية في الشبكة grid" ،وتضع الخاصية DescriptionAttributeالنص الذي يظهر في منطقة المساعدة(أو التعليمات)في الأسفل للأداة PropertyGrid.

عندما استخدمت لوحة الخاصيات لتحرير نموذجك،فقد كنت قادر على وضع ألوان من أجل خاصية اللون باستخدام أداة اختيار لون خاصة مبنية ضمن الخاصية. مجرد أن يكون لديك خاصية معرفة باستخدام System.Drawing.Color فهذا كافي لتمكين نفس التخصص الوظيفي من أجل فئتك الخاصة.ولكن كيف يعمل هذا؟مجرد أن يكون للخاصية FillColor مواصفات يتم تميزها بواسطة الأداةPropertyGrid، ولدى الفئة System.Drawing.Color أيضاً مثل هذه الخاصيات،واحد منها تعرف فئة محرر خاصية خاصة من الألوان.وتنفيذها يقع خلف مجال هذا الكتاب، وعلى أية حال جامدة.إذا كنت مهتم بمثل هذا العمل من أجل فئاتك الخاصة،تستطيع قراءة مواضيع بخصوص محررات شبكة خاصية من مستندات ميكروسوفت.

قبل أن ندخل إلى نماذج التحرير ،سأتيح لك معرفة شيء ما يخص دعم الدوال التي أضفتها منذ حين إلى ملف الوحدة البرمجية *General.vb*.

الدالة:BuildFontStyle

أنماط الخط(مثل غامق،ومائل)هي مجموعة في كائنات الخط تستخدم أعضاء العداد System.Drawing.FontStyle.ولكن عند تخزين معلومات خط في قاعدة البيانات،اخترت تخزين إعدادات الأنماط هذه باستخدام أحرف(مثل B من أجل الغامقbold).تعمل هذه الدالة على تحويل الحرف إلى قيمة نمط الخطFontStyle .

الدالة: ConvertPageUnits

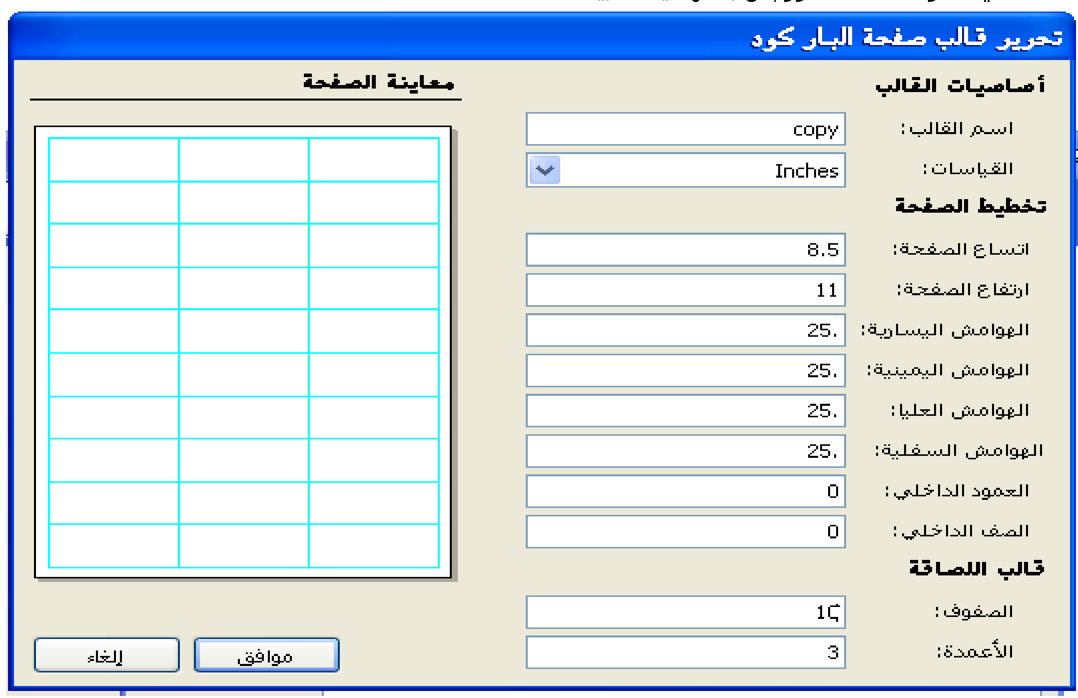
يتيح لك محرر اللصاقة label editors وضع البنود في عدة أنظمة قياس مختلفة، متضمنة الانش والسنتيمتر.تحول هذه الدالة القياسات بين الأنظمة المختلفة.

الدالة:DBFontStyle

هذه الدالة عكس الدالة BuildFontStyle،تحضر قيمة نمط خط FontStyleمن أجل إدخالها في سجل قاعدة البيانات.

الدالة: GetBarcodeFont

تعود هذه الدالة باسم خط البار كود،إذا تم تركيبه. تسمح الفورم للمستخدم من تعريف صفحة كاملة للصاقة.وليس اللصقات نفسها ولكن مواضع لصاقات متعددة على نفس صفحة الطباعة.يبين الشكل التالي الحقول على هذه الفورم مع بعض عينات البيانات.



إجمالاً،تشرح الحقول على الفورم حجم الصفحة وحجم كل لصاقة تظهر على الصفحة.عندما يعمل المستخدم على إدخال القيم،فإن منطقة معاينة الصفحة تتجدد مع المعاينة لما ستبدو عليه الصفحة.بما أن الكود مشتق من الفورمBaseCodeForm، فإن المنطق في الفورم معروف لك مسبقاً،إنه يدير البيانات الموجودة في سجل واحد من الجدول BarcodeSheet. ما هو مختلف هو كود +GDIالموجود في معالج الحدثPreviewArea.Paint. مقطع الكود الأول الرئيسي يحاول تحديد كيفية موازنة قطعة 8.5 · 11من الصفحة لجعلها تظهر في مستطيل صغير حجمه216·272.وهناك الكثير من الحسابات،عندما تكتمل،يتم تحديد نسبة الصفحة الصغيرة إلى الكبيرة،ويؤدي إلى رسم قطعة من الصفحة على الشاشة مع الحدود وظل الإسقاط

```
' رسم منطقة الصفحة
    e.Graphics.FillRectangle(SystemBrushes.ControlDark, pageLeft + 1, _
        pageTop + 1, pageWidth + 2, pageHeight + 2)
    e.Graphics.FillRectangle(SystemBrushes.ControlDark, pageLeft + 2, _
        pageTop + 2, pageWidth + 2, pageHeight + 2)
    e.Graphics.FillRectangle(Brushes.Black, pageLeft - 1, pageTop - 1, _
        pageWidth + 2, pageHeight + 2)
    e.Graphics.FillRectangle(Brushes.White, pageLeft, pageTop, pageWidth, pageHeight)
```

ومن ثم وقبل رسم الحدود الخارجية للمعاينة بالنسبة لكل لصاقة مستطيلة،تعمل على إعادة موضعة مصدر الشبكة إلى الزاوية العلوية اليسارية بالنسبة للقطعة التي على الشاشة من الورقة،وتحويلات ميزان الكلمة بالاعتماد على نسبة قطعة الكلمة الحقيقية للورقة والصورة على الشاشة لها.

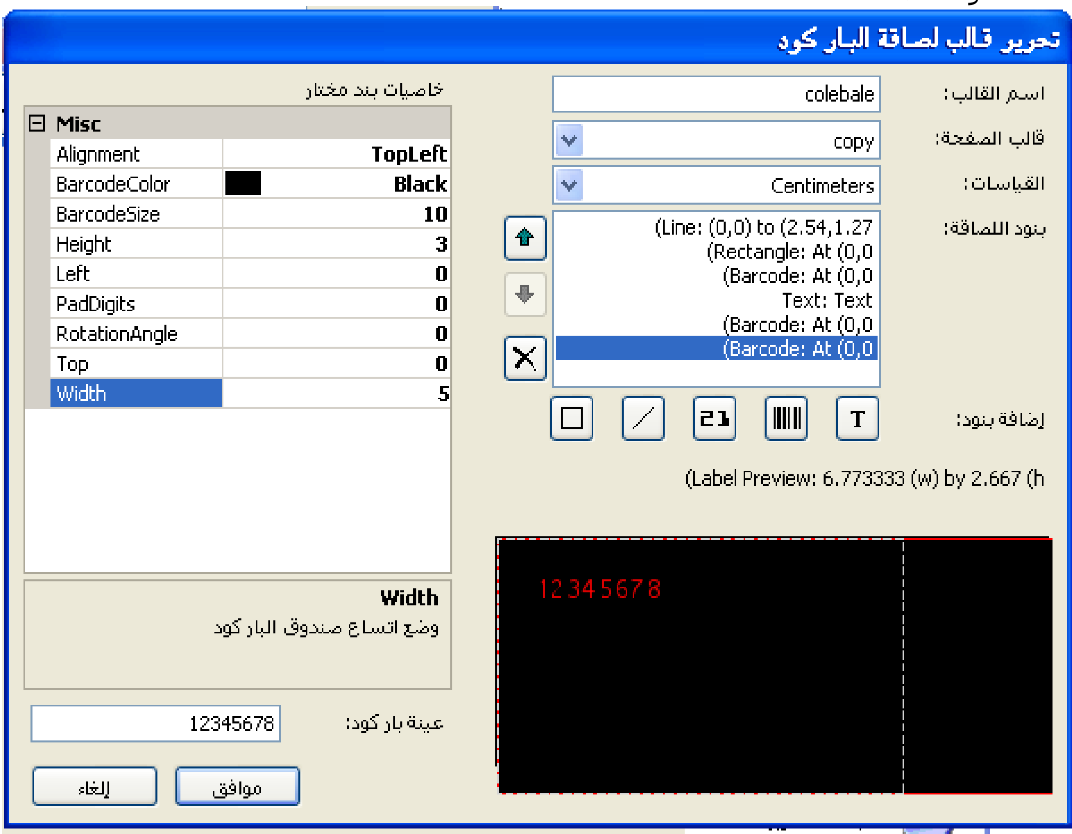
```
e.Graphics.TranslateTransform(pageLeft, pageTop)
```

```
e.Graphics.ScaleTransform(useRatio, useRatio)
```

توجد العديد من الحسابات الأخرى من أجل الحجم لكل لصاقة،متبوعة بحلقة مضاعفة(من أجل كل من الصفوف والأعمدة للصاقة)وهي تقوم بالطباعة الحقيقية لحدود اللصاقة (تفاصيل الحسابات تم حذفها من أجل الاختزال).

```
For rowScan = 1 To CInt(BCRows.Text)
            For colScan = 1 To CInt(BCColumns.Text)
leftOffset = ...
topOffset = ...
                e.Graphics.DrawRectangle(Pens.Cyan, _
                leftOffset, topOffset, _
                oneWidthTwips, oneHeightTwips)
            Next colScan
Next rowScan
```

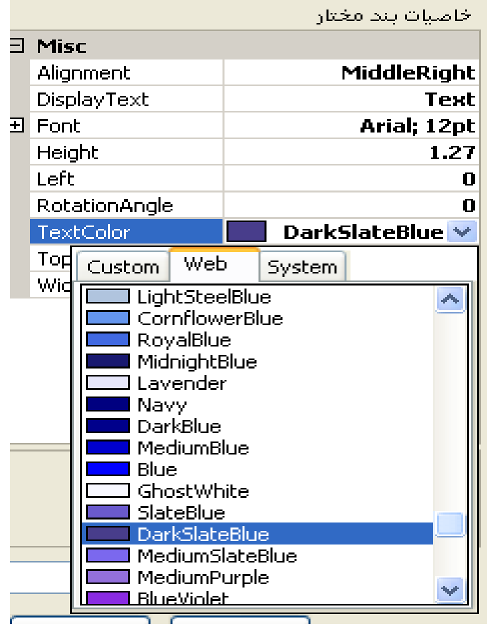
من الواضح أن الفورم BarcodeLabel أكثر أهمية وتعقيداً من نموذجي تحرير البار كود.بينما يعرف النموذج BarcodePage الصفيحة الكاملة للصاقات مع لاشيء بداخل كل لصاقة،تعرف الفورم BarcodeLabel ما يجري داخل كل لصاقة،يبين الشكل التالي هذه الفورم مع عينة لصاقة .
إن الفورم BarcodeLabel أيضاً مشتقة من الفورم BaseCodeForm، الكثير من كودها يتعامل مع تحميل وحفظ السجلات من جداول قاعدة البيانات BarcodeLabel وBarcodeLabelItem. كل لصاقة بار كود مربوطة إلى قالب صفحة بار كود خاصة(والتي حددناها منذ قليل من خلال الفورم BarcodePage)،وتخزين سجلها الرئيسي في الجدول BarcodeLabel. يحدد هذه الجدول أساسيات اللصاقة،مثل اسمها ونظام القياسات.نص وشكل بند موضوع على الجدول يتم تخزينهما كسجلات في الجدول BarcodeLabelItem ذو الصلة.



يعمل الروتين PrepareFormFields على تحميل سجلات لصاقة سجلات موجودة من قاعدة البيانات،منشأ حالة فئات من الملف *BarcodeItemClass.vb* الجديد،ويضيفها إلى أداة صندوق القائمة DisplayItems.إليك مقطع الكود الذي يحمل في "صورة البار كود"(البار كود الفعلي المعروض)من مدخلة في جدول BarcodeLabelItems.

```
'صورة البار كود
newBarcodeImage = New Library.BarcodeItemBarcodeImage
newBarcodeImage.Alignment = CType(CInt(dbInfo!Alignment), System.Drawing.ContentAlignment)
newBarcodeImage.BarcodeColor =System.Drawing.Color.FromArgb(CInt(dbInfo!Color1))
                    newBarcodeImage.BarcodeSize = CSng(dbInfo!FontSize)
                    newBarcodeImage.Left = CSng(dbInfo!PosLeft)
                    newBarcodeImage.Top = CSng(dbInfo!PosTop)
                    newBarcodeImage.Width = CSng(dbInfo!PosWidth)
                    newBarcodeImage.Height = CSng(dbInfo!PosHeight)
                    newBarcodeImage.RotationAngle = CShort(dbInfo!Rotation)
                    newBarcodeImage.PadDigits = CByte(DBGetInteger(dbInfo!PadDigits))
                    DisplayItems.Items.Add(newBarcodeImage)
```

بإمكان المستخدم إضافة أشكال،عناصر نص،وبار كود إلى اللصاقة بالنقر على واحدة من أزرار إضافة بند التي تظهر تحت أداة DisplayItems.كل زر يعمل على إضافة سجل افتراضي إلى اللصاقة،بحيث يستطيع المستخدم تعديله فيما بعد.عند يتم اختيار كل عنصر لصاقة من DisplayItems،تظهر خاصياته في الأداة ItemProperties،وهي حالة أو نسخة عن الأداة PropertyGrid.تعديل عنصر لصاقة هي قضية تغير خاصياته.يبين الشكل التالي تغير خاصية اللون.



كما مع الفورمBarcodePage، تأتي المتعة الحقيقة في الفورم BarcodeLabelمن خلال حدث الرسم Paintلأداة معاينة previewاللصاقة، PreviewArea،وهذا الروتين حوالي 300سطر يبدأ برسم السطح الفارغ blank surface للصاقة مع إسقاط الظل.ومن ثم يعالج كل عنصر في القائمةDisplayItems، واحد بعد الآخر،بالتالي يعمل على تحويل ورسم كل عنصر كما تشير خاصياته.عند تمرير الخاصيات من خلال قائمة عنصر، يطبق الكود التحويلات إلى منطقة الرسم عند الحاجة.ولحفظ الأشياء مرتبة من أجل كل عنصر،يتم حفظ بداية السطح قبل عمل التغيرات،ويتم إتمام التغيرات المعاد تخزينها.

```
'معالجة كل بند في القائمة
        For counter = 0 To DisplayItems.Items.Count - 1
'          حفظ الحالة الحالية لمنطقة الرسم
            holdState = e.Graphics.Save()
```
' كود الرسم الرئيسي يتم وضعه هنا.
```
'تجديد الحالة المحولة الأصلية لمنطقة الرسوميات
            e.Graphics.Restore(holdState)
Next counter
```

ينجز كل كود عنصر حجوم متنوع،موضع،وتحويلات تدوير مطلوبة من أجل العرض المناسب للعنصر.لنلقي نظرة أقرب على الكود الذي يعرض عناصر النص الثابت(الكود الذي يتم استدعاءه لعرض نص البار كود). بعد ترتيب عرض الكلمة إلى منطقة معاينة سطح اللصاقة،يتم عمل أي تدوير مطلوب من قبل المستخدم بخصوص الزاوية العليا اليسارية للمستطيل الذي يحفظ نص الطباعة.

```
e.Graphics.TranslateTransform(X1, Y1)
e.Graphics.RotateTransform(textAngle)
```

التالي،رسم سطر متقطع حول كائن النص لإظهار الحالة المختارة.

```
pixelPen = New System.Drawing.Pen(Color.LightGray,1 / e.Graphics.DpiX)
  pixelPen.DashStyle = Drawing2D.DashStyle.Dash
  e.Graphics.DrawRectangle(pixelPen, X1, Y1, X2, Y2)
pixelPen.Dispose()
```

بعد إعداد بعض علامات تجانب النص المناسب عمودياً وافقياً ضمن حدود صندوقها،تدفع الطريقة DrawStringالنص إلى العرض.

```
  e.Graphics.DrawString(textMessage, useFont,New System.Drawing.SolidBrush(textColor), _
                      New Drawing.RectangleF(X1, Y1, X2, Y2), textFormat)
  useFont.Dispose()
```

سنعمل على تكرار كود رسم اللصاقة المضمن في الفئة BarcodeLabel نوعاً ما عندما نطبع اللصاقات الحقيقية في فصل لاحق.
الشيء الوحيد الباقي هو ربط هذه المحررات مع الفورم الرئيسي.افتح كود الفورم الرئيسيةMainForm، ابحث عن معالج حدث AdminLinkBarcodeLabel.LinkClicked،وأضف له الكود التالي:

```
Private Sub AdminLinkBarcodeLabel_LinkClicked(ByVal sender As Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles AdminLinkBarcodeLabel.LinkClicked
'           دع المستخدم يحرر قائمة لصاقات البار كود
        If (SecurityProfile(LibrarySecurity.ManageBarcodeTemplates) = False) Then
            MsgBox(NotAuthorizedMessage, MsgBoxStyle.OkOnly Or _
                MsgBoxStyle.Exclamation, ProgramTitle)
            Return
        End If
'           تحرير السجلات
```

```
        ListEditRecords.ManageRecords(New Library.BarcodeLabel)
        ListEditRecords = Nothing
    End Sub
```

افعل المثل مع معالج حدثAdminLinkBarcodePage.LinkClicked. فكوده مطابق لهذا الكود ما عدا حالة أو نسخة الفئة إلى ListEditRecords.

```
Private Sub AdminLinkBarcodePage_LinkClicked(ByVal sender As Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles AdminLinkBarcodePage.LinkClicked
        ' دع المستخدم يحرر قائمة صفحات البار كود
        If (SecurityProfile(LibrarySecurity.ManageBarcodeTemplates) = False) Then
            MsgBox(NotAuthorizedMessage, MsgBoxStyle.OkOnly Or _
                MsgBoxStyle.Exclamation, ProgramTitle)
            Return
        End If
        ' تحرير السجلات
        ListEditRecords.ManageRecords(New Library.BarcodePage)
        ListEditRecords = Nothing
    End Sub
```

المتعة بالرسم.Fun with Graphics

ليس كل الـ يخص أشياء الرسم الحقيقية،تستطيع أن تحصل على بعض المتعة.دعنا نعمل تغيرات على الفورم"حول المشروع *AboutProgram.vb*" والتي تتلاشى عندما ينقر المستخدم على الزر إغلاقClose. وهذا يتضمن تبديل خاصية الغباش Opacityبحيث تزداد الشفافية transparencyببطء بالنسبة للفورم.من وجهة نظر كودك،فليس هناك +GDIمضمن.ولكن ما يزال مضمن من خلال الكود المخفي والذي يستجيب على خاصية الغباش(أو اللاشفافيةOpacity).افتح الكود المصدري لهذه الفورم،وأضف الكود التالي إلى نهاية معالج حدث تحميل الفورم AboutProgram.Load.

على الرغم من أن هذه العبارة ضرورية،وجدت أن الفورم تميل لأن تومض قليلاً على الأنظمة عندما تنتقل اللاشفافية من 100%(1.0)إلى أي شيء أخر(99%أو 0.99 ،في هذه الحالة).هذه الومضة كانت أقل ملاحظةً عندما عملت المقطع الانتقالي أثناء معالجة التحميل.

في معالج حدث نقر الزر إغلاقActClose.Click، اعمل على تضمين هذا الكود :

```
'تلاشي الفورم
        Dim counter As Integer
        For counter = 90 To 10 Step -20
            Me.Opacity = counter / 100
            Me.Refresh()
            Threading.Thread.Sleep(50)
        Next counter
        Me.DialogResult = Windows.Forms.DialogResult.Cancel
```

يعمل هذا الكود على إغلاق الفورم بشكل متلاشي خلال ميلي ثانية،في خمس خطوات متميزة.لذلك فإن الفورم لايتم إغلاقها بشكل مفاجئ قبل التلاشي ،افتح مصمم الفورم،اختر الزر "إغلاق ActClose" وغير خاصيته DialogResultإلى None.

شيء أخر لم نعمله أبداً وهو وضع الأيقونة الرئيسية للتطبيق.على الرغم من هذا ليس بالضبط،فإنه يتضمن عرض غرافيك،والتي تؤثر بإدراك أو تقييم المستخدم لنوعية البرنامج.عملت على تضمين أيقونة مسماة *Book.ico* في مجموعة ملف المشروع.افتح خاصيات المشروع،اختر تبويب تطبيقApplication، واستخدم حقل الأيقونة Iconللتصفح والبحث عن ملف *Book.ico*.

أثناء تجريب الأيقونة،لاحظت أن النافذة المنتشرة تظهر(مع أيقونة الفيجوال أستوديو الافتراضية) في شريط مهام ويندوز.في الحقيقة،كل فورم مفتوحة تظهر في شريط المهام،على يسار مدخلة الفورم الرئيسي.وهذا ليس قياسي،وكل هذا يعود إلى الخاصية ShowInTaskbarالمضمنة مع كل فورم،فعملت على وضع هذه الخاصية إلى خطأ ما عدا الفورم الرئيسية MainForm.

مع الفصل التالي سنكون قد عملنا على إضافة أكثر من 95% من الكود الرئيسي للمشروع.

الحصر والشمولية Localization and Globalization

بخصوص محاولة توسيع جاذبية appealتطبيقاتك الخاصة لما خلف العالم المتحدث الانكليزية،توفر الدوت نت ميزات تتيح لك تخصيص *localize*تطبيقك في لغة أخرى،حتى بعد أن يتم ترجمة تطبيقك ونشره.
تغطية جميع ميزات التخصيص في الدوت نت سيتضمن تقويمات الفترة الإمبراطورية والإسلامية،وأنظمة الكتابة من اليمين إلى اليسار.يغطي هذا الفصل ميزات تخصيص واجهة المستخدم الأكثر شيوعاً.على أمل،أن تغريك في توسيع حدود اللغة.

تعريف الشمولية والتخصيص(الحصر أو المركزة).Defining Globalization and Localization

إن لدى ميكروسوفت المئات من تطبيقات البرمجيات المتاحة للبيع والمجانية،وتجمع الشركة الكثير من النقود على امتداد العالم من خلال توفير منتجات برمجية للزبائن.
معظم منتجاتها تم تطويرها في الولايات المتحدة،وتم كتابتها من قبل مبرمجين يتحدثون بشكل رئيسي اللغة الإنكليزية،وتم إخراجها بواسطة قيادات تقنية ومديري منتجات يصنعون القرارات بالإنكليزية،ويتم تسويقها بواسطة فريق مبيعات والذي يخطط مسبقاً للحملات التجارية بالإنكليزية،ويتم مضايقتها doggedمن قبل المنافسين competitorsوالمنتقدين detractorsالذين ينسفون blast الحوافز motivesومزاولة الأعمال خلف كل منتج بالإنكليزية.لذلك،ما هو متاح هو أن بإمكان ميكروسوفت بيع البرمجيات إلى غير الناطقين بالانكليزية حول المعمورة.
يكمن المفتاح في شمولية وتخصيص منتجاتها.يمكن لميكروسوفت أو أية شركة أخرى أن تطور منتجات متماثلة متميزة،وحتى بلغات مختلفة،وتبيعها في أسواق كافية.ولكن سيكون ذلك مكلفاً ومضيعة للوقت.البديل،كتابة برنامج مفرد،ومن ثم تحسينه من خلال ميزات تعين اللغة والثقافة.
التعميم(الشمولية)*Globalization* هو عملية تحضير برمجيات بحيث يمكن ضبطها بسهولة بالنسبة لكل سوق ثقافي ولغة.لا يتم إضافة مصطلحات غريبة إلى البرمجيات خلال عملية التعميم. بدلاً عنه،يعمل المطورين على تصميم التطبيق بحيث أن كل ما يتعلق بمصطلحات الإنكليزية والعناصر الثقافية الأمريكية(مثل عرض العملة بالدولار الأمريكي)يمكن أن يتم بسرعة وسهولة تبديلها جميعاً بواسطة البديل الأجنبي دون أن يؤثر على عناصر البرمجيات الأساسية.
تعمل تطبيقات ويندوز بشكل تقليدي على استخدام"الموارد" لحفظ التطبيق بشكل عام شمولي.تحتوي الموارد سلاسل نصية،صور،وعناصر أخرى غير الكود يتم استبدالها وقت التنفيذ بالاعتماد على اللغة النشيطة(الفعالة) والثقافة(الإعدادات الإقليمية) لنظام التشغيل.
ففي نظام باللغة الألمانية،يعمل التطبيق على تحميل موارده باللغة الألمانية (إذا كانت متاحة)ويعرضها عوضاً عن الموارد الافتراضية.يواصل إطار عمل الدوت نت استخدام الموارد من أجل هذا الهدف،على الرغم من أنه يحسن تطوير الموارد من خلال ملفات موارد معتمدة على XML وأدوات.
التخصيص(الحصر) *Localization* يعمل على إضافة اللغة الغير أصلية الفعلية وعناصر الثقافة إلى التطبيق.إنه يقول في هذه الخطوة، اعمل على ترجمة لافتات فورم اللغة الإنكليزية إلى أية لغة أخرى مثل اللغة العربية.تتيح لك الفيجوال أستوديو تخصيص(مركزة)التطبيق ضمن بيئة التطوير نفسها،أو من خلال أدوات خارجية حيث أن المترجمات ليس لديها صلاحية الوصول إلى الكود المصدري للتطبيق لتتمكن من استخدامه.
الأخبار الحسنة بالنسبة لمطوري الدوت نت هو أن ميكروسوفت أخذت على عاتقها الكثير من الاهتمام بجزء التعميمglobalization .بشكل رئيسي تحتاج أن تركز على تخصيص تطبيقك.مجموعات جامعاتك المحلية توفر تعليم للغات الأجنبية بالكثير من اللغات،لذلك سأترك لك حرية اختيار تخصيص اللغة الهدف.

ملفات الموارد.Resource Files

ملفات الموارد هي المفتاح إلى لغة التخصيص في برامج الدوت نت.ستكتب الفيجوال أستوديو الملفات عوضاً عنك،ولكن من الجيد معرفة شيء ما حول كيفية عملها،بما أنك تريد صناعة ملفات الموارد الخاصة بك(إذا كان لديك الكثير من الوقت).تسير حياة مصدر ما في ثلاث أطوار،كما تم تحديدها من قبل نوع الملف الذي تظهر فيه.

المصدر.*Source*

تبدأ موارد التطبيق حياتها في ملف مصدر الموارد.قبل الدوت نت،كانت الموارد تظهر في ملفات"صيغة المورد resource script "،والتي تدمج أفضل مافي لغة التطوير Cو أوامر صيغة الحالة الكبيرة ،وتستخدم امتداد الملف *.rc*. في الفيجوال بيسك2008،تستخدم XML معتمد على ملفات *.resx*. يتضمن كل تطبيق نماذج ويندوز جديد ملف *Resources.resx*،ينتظر أن تعمل على تعبئته بمصادر تطبيقك.
خلف ملفات مصدر الموارد الأساسية،يمكن أن يتم تضمين أنواع ملفات أخرى كموارد،على الرغم من أنها ستبقى مراجع من خلال محتوى الملف*resx*.. تتضمن ملفات الموارد الخارجية المشتركة ملفات صورة(مثل ملفات*gif*. و *jpg*.) وملفات نصية (*txt*.)).يستخدم مشروع المكتبة ملف اسمه *SplashImage.jpg* كمورد من أجل الشاشة المنتشرة،وملف أخر مسمى *ItemLookupBody.txt*يحتوي محتوى HTMLيتم استخدامه عند عرض بنود من خلال الفورم *ItemLookup.vb*.

الوسيط.*Intermediate*

حالما تكون مصادر مواردك جاهزة،يتم تحويلها إلى نموذج وسيطintermediate ،ويتم تخزينها مع امتداد ملف الموارد*resources*.،من خلال معالجة تدعى إنتاج المورد.تعمل عادة الفيجوال أستوديو هذه الخطوة خلف الكواليس من أجلك،ولكن تستطيع أيضاً استخدام أداة موفرة من قبل مجموعة تطوير برمجيات الدوت نتNET SDK. (تدعى *resgen.exe*)لإنتاج هذه الملفات بنفسك.تتضمن ملفات المورد الوسيطة محتوى ثنائي فقط،ولم يتم تصميمها من أجل الاستعراض في المفكرة.

المترجم.*Compiled*

ملفات المورد الوسيطة لايتم استخدامها كثيراً من أجل توزيع تطبيقك.فالمصطلح"وسيط *intermediate* "من النوع الذي يتخلى عن الخصوصية أو يطرح الأسرار جانباً،قبل توزيع مواردك في البرنامج،تحتاج إلى أن يتم ترجمتها إلى ملف DLLأو EXE.من المحتمل أنك تعلم مسبقاً أن هذه الملفات تحتوي عدة مقاطع،متضمنة كود متميز ومقاطع بيانات.يحتوي ملف الموارد المترجم مقطع البيانات فقط مع الموارد،فلا يوجد كود في ملف المورد المترجم،على الرغم من ذلك يمكن أن تتضمن ملفات الكود المترجمة القياسية أيضاً مصادر مترجمة.
في الدوت نت،ملفات الموارد المترجمة هي مجمعات تابعة.تدعم مجمع التطبيق الرئيسي،وهي غير مفيدة بشكل عام بعيداً عن المجمع الرئيسي.
بعض أنواع الموارد القياسية يتم تخزينها في ملفات الموارد(*resx*.) للدوت نت كــ :

سلاسل حرفية.*Strings*

سنركز بشكل رئيسي على موارد السلاسل الحرفية في هذه الفصل.كل مورد سلسلة حرفية يتضمن اسم وقيمة نصية.

الصور.*Images*

يمكن أن تتضمن تطبيقات الفيجوال بيسك ملفات الصور JPEG، GIF، TIFF، PNG،و BMP.كل صورة،كما جميع الموارد،تتضمن الاسم المرافق،والذي يمكن أن يختلف عن الاسم الأصلي لملف الغرافيك(أو الرسوميات).

الأيقونات*Icons*

أيقونات البرنامج المستخدمة مع النماذج والتطبيق نفسه تظهر كموارد قياسية. للأيقونات امتداد ملف*ico.* .

الأديو*Audio*

يمكن أن تتضمن الموارد ملفات أوديو محددة،بالاعتماد على محتوى أوديو ويف WAV.

ملفات*Files*

إذا كانت أنواع الملفات المجدولة حتى الآن لا تلبي احتياجاتك،تستطيع تضمين جميع الملفات لأي نوع كموارد مسماة.

أخرى*Other*

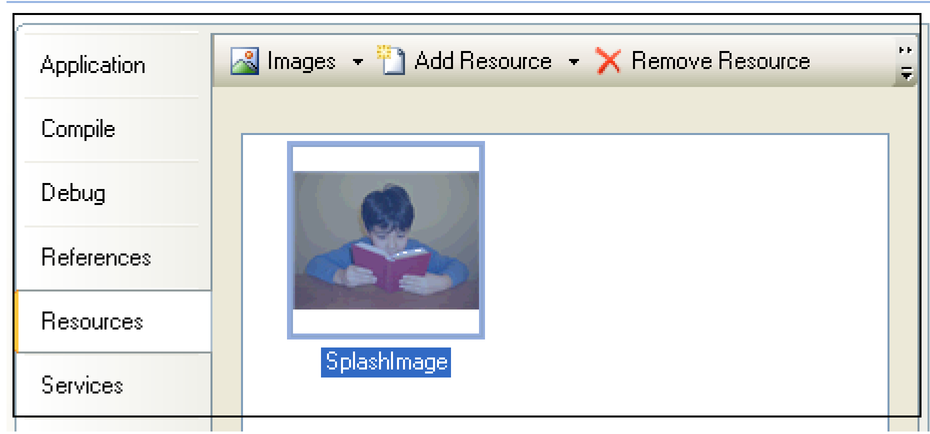
خلف الملفات،تستطيع تخزين محتوى أي نوع بيانات دوت نت كمورد.الموارد في ملف*resx.* هي محددة النوع بقوة بالنسبة لأنواع الدوت نت،لذلك لاتوجد حدود حقيقية لنوع البيانات التي تستطيع وضعها هناك.تستطيع أيضاً تعديل محتوى الملف *resx.* بحيث يتضمن موارد غير قياسية. وتقع الموارد غير القياسية non-standard خلف حدود هذه الفصل.

تتضمن نافذة خاصيات المشروع مدير من أجل موارد التطبيق الواسعة(شاهد الشكل التالي).تتضمن بيئة التطوير المتكاملة أيضاً محررات خاصة تتيح لك تحرير أنواع الموارد القياسية والعديد من أنواع الموارد الغير قياسية.

الكائن My.Resources.

ناقشنا هذه الكائن في فصول سابقة،ولكن كتذكير،تستطيع الوصول إلى موارد التطبيق من خلال الكائنMy.Resources. إذا كان لديك مورد نصي مسمىMainFormCaption ، فإن المرجع التالي يعود بقيمته:`My.Resources.MainFormCaption`

جميع الموارد محددة النوع بقوة.في هذه الحالة، MainFormCaptionمن نوع System.String. مورد الصورة SplashImageالمضمنة في مشروع المكتبة يتم التصريح عنها كنوع System.Drawing.Bitmap.لأن كل مورد محدد النوع بقوة،تستطيع استخدام المرجع My.Resourcesفي كودك مثل أي بيانات نوع مورد تماماً.



في تطبيقات نماذج ويندوز الجديدة،تظهر جميع موارد التطبيق الواسعة في الملف *Resources.resx*،تجدها في الدليل *My Project* ضمن دليل الكود المصدري للتطبيق، تستطيع عرضها في المفكرة إذا أردت،وهي ملف XMLكبير نسبياً ،ولاتهمني مباشرةً،ما عدا عملها،إليك قسم من ملف*Resources.resx* لمشروع المكتبة والذي يخصص الموردين الموجودين فيه(قمت بحذف بعض الأسطر لجعله يناسب الصفحة)ولقد علمت اسم كل مورد،وأنواع بياناتها المحددة بقوة.

```
<data name="ItemLookupBody" type="System.Resources.ResXFileRef, System.Windows.Forms">
    <value>..\Resources\ItemLookupBody.txt;System.String, mscorlib, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089;Windows-1252</value>
  </data>
  <data name="SplashImage" type="System.Resources.ResXFileRef, System.Windows.Forms">
    <value>..\Resources\SplashImage.jpg;System.Drawing.Bitmap, System.Drawing,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a</value>
  </data>
```

كل فورم تضيفه إلى المشروع لديه أيضاً ملف المورد الخاص به.فمن أجل الـ Form1يدعى*Form1.resx*. هذه الملفات ينتهي بها المطاف لأن تكون إضافة كبيرة في مركز تطبيقات نماذج ويندوز.

خلف الستار،يأخذ تطبيقك محاكاة التوجه الكائني لإدارة الموارد.فهو يستخدم الفئة System.Resources.ResourceManager لإيجاد والعودة بحالات كل مورد عندما تحتاجه.ونفس هذه الفئة تصنع القرار بخصوص تحديد لغة معينة أو الموارد الخاصة بثقافة من بين الكثير من الموارد المضافة لتطبيقك وستجعلها مرئية للمستخدم.

تخصيص النماذج ضمن الفيجوال أستوديو.Localizing Forms Within Visual Studio

لايوجد معنى لتأجيل تقديم ميزات التخصيص في الفيجوال أستوديو،بما أنها سهلة الاستخدام.لقد علمت مسبقاً ما يخص محرر موارد خاصيات المشروع الواسعة للتطبيق. بالمقابل، لنلقي نظرة على جزء مميز:تخصيص النماذج والأدوات الموجودة في محرر نماذج الفيجوال أستوديو.

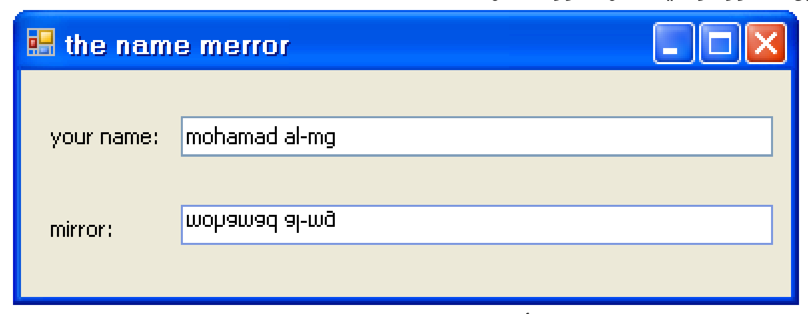
إليك تطبيق نماذج ويندوز يعمل على كتابة اسمك بشكل مقلوب،عملت على إضافة أدوات لصاقة Label،وأداة صندوق نصTextBox، وصندوق صورة PictureBox كما هو مبين في الشكل التالي.



ومن اعمل على إضافة الكود المصدري التالي إلى الفورم:

```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged
إعادة الرسم '
        PictureBox1.Invalidate()
    End Sub
    Private Sub PictureBox1_Paint(ByVal sender As Object, ByVal e As System.Windows.Forms.PaintEventArgs) Handles PictureBox1.Paint
رسم الخلفية الفارغة '
        e.Graphics.Clear(SystemColors.Window)
        e.Graphics.DrawRectangle(SystemPens.InactiveCaption, 0, 0, PictureBox1.Width - 1, PictureBox1.Height - 1)
تغير اتجاه العرض '
        Dim saveState As Drawing2D.GraphicsState = e.Graphics.Save()
        Dim mirrorMatrix As New Drawing2D.Matrix(1, 0, 0, -1, 0, PictureBox1.Height)
        e.Graphics.Transform = mirrorMatrix
رسم النص '
        e.Graphics.DrawString(TextBox1.Text, TextBox1.Font, SystemBrushes.WindowText, 1, 4)
إعادة كل شيء '
        e.Graphics.Restore(saveState)
    End Sub
```

عندما تشغل البرنامج،فإنه يعمل على إنشاء صورة عاكسة لما تكتبه في أداة صندوق النص باستخدام ميزات الـ +GDI،يبين الشكل التالي تشغيل التطبيق وكتابة بعض النص في صندوق النص وكيف يظهر في أداة صندوق الصورة والتي تعكس صورة النص.



بقدر أهمية هذا البرنامج فهو ليس معمم بالكامل ولا حتى مخصص.وعلى الأغلب شامل.كل ما نحتاج عمله لنجعله معمم بالكامل"إطلاق التحويل"على الفورم الذي يمكّن فيما بعد التخصيص.إننا نعمل هذا من خلال خاصية الفورمLocalizable. غير هذه الخاصية من خطأ Falseإلى صوابTrue. هل فورمك معممة!.

الآن من أجل الجزء الثاني:التخصيصlocalization. إليك الخطوات من أجل تخصيص الفورم:

1.حدد اللغة أو ضم لغة الثقافة التي تريد تخصيصها.

2.اختر تلك اللغة أو لغة الثقافة من خاصية النموذج Language.

عندما تفتح قائمة هذه الخاصية،فهي تتضمن اللغات الموجودة على نظامك،مثل الفرنسية،العربية،واللغات المضمومة مع الثقافة والبلد.واللغة التي تخصصها هي التي يستطيع المستخدم استخدامها.

3.عدل أي خاصية للفورم أو أدواتها.

وهذا يتم عندما يتم تغير خاصية اللغة للفورم إلى لغة غير الافتراضيةDefault، فإن الفيجوال أستوديو تبدأ بتسجيل جميع تغيرات الفورم والأدوات إلى ملف موارد خاص بالثقافة أو اللغة منفصل بالنسبة للفورم.

تستطيع تخصيص الفورم مع عدة لغات.كل مرة تغير خاصية اللغة إلى لغة أخرى أو اختيار لغة الثقافة،يتم تطبيق التغيرات للفورم والأدوات فقط إلى ذلك الاختيار.ما تغيره يتم حفظه في ملف موارد منفصل.

دعنا نجربه مع البرنامج الحالي،سأعمل على اختيار لغة تخصيص للعربية.أولاً،سأعمل على وضع خاصية اللغة للفورم Language إلى العربية.تومض الفورم بشكل لحظي،ولكن ليس هناك ملاحظات أخرى. فهي تبدو كما في الشكل في الأعلى.

التالي سأعمل على تغير خاصية النص للفورم Textوكل أداة لصاقة إلى اللغة العربية المكافئة مع جعل خاصيات التخطيط من اليمين إلى اليسار والخاصية من اليمين إلى اليسار إلى صواب فستبدو الفورم كما هو مبين في الشكل التالي.



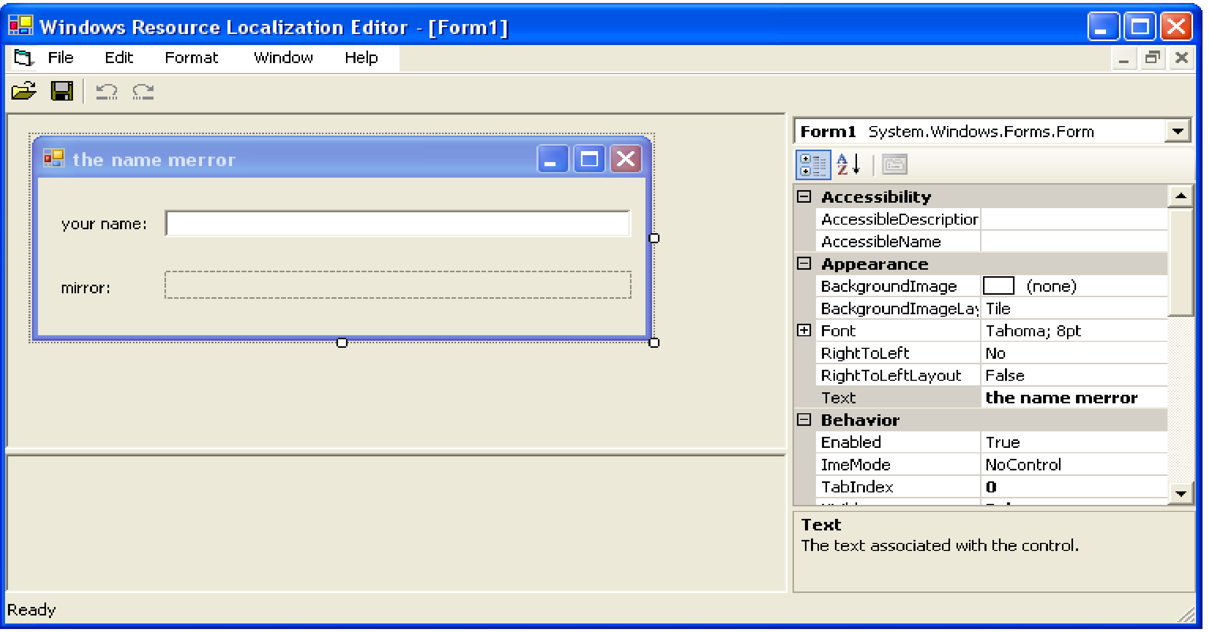
الجزء المدهش،إذا ما عملت على إعادة وضع خاصية اللغة Languageللفورم مرةً أخرى إلى (Default)ليست اللصاقات فقط تعود إلى اللغة الإنكليزية،ولكن الحقول المقابلة لها ستعود أيضاً إلى ما كانت عليه،على الرغم من أنني لم أراجع كل خاصية،ولكن على ما يبدو أن خاصية التخصيص تؤثر على جميع عناصر العرض لكل أداة.
إن البرنامج الآن مخصص بالكامل إلى اللغة الإنكليزية(اللغة الافتراضية)،واللغة العربية .
عادةً،الموارد العربية سيتم استخدامها فقط على أنظمة تشغيل باللغة العربية لميكروسوفت ويندوز،ولكن نستطيع إجبار البرنامج على استخدام اللغة العربية وذلك بتغير "ثقافة واجهة المستخدم" في كود بدء التطبيق (الروتين) MyApplication_Startup في الملف*ApplicationEvents.vb* (سأعمل على إضافة الكود التالي:

```
Private Sub MyApplication_Startup(ByVal sender As Object, ByVal e As
Microsoft.VisualBasic.ApplicationServices.StartupEventArgs) Handles Me.Startup
            If (MsgBox("التبديل من الإنكليزية إلى العربية؟", MsgBoxStyle.Question Or MsgBoxStyle.YesNo) =
MsgBoxResult.Yes) Then
                My.Application.ChangeUICulture("AR")
            End If
        End Sub
```

شغل البرنامج الآن وجربه،ولكن يجب أن يكون نظام التشغيل لديك باللغة الإنكليزية لكي يتم التبديل بين العربية والإنكليزية وإلا إذا كان نظام تشغيلك باللغة العربية فإن الافتراضي أن يعمل هذا البرنامج باللغة العربية بشكل افتراضي لأنها لغة نظام التشغيل الافتراضية.ولكن إذا كان باللغة الإنكليزية فإن يتم التبديل بين اللغتين.

إضافة موارد من خارج الفيجوال أستوديو.

تجعل الفيجوال أستوديو التخصيص سهل جداً.ولكن من النادر أن يكون مطور تطبيق رئيسي طليق اللسان في عدة لغات.وبالتأكيد لاتريد من غير المبرمجين من أن يتمكنوا من الوصول إلى نموذجك والكود في الفيجوال أستوديو،حيث يمكنهم أن يعملوا ما يعرف بالمنطق.
لحفظ عيون وأصابع اللغة الأجنبية إلى ما تنتمي،فقد كتبت ميكروسوفت محرر تخصيص موارد ويندوز،وضمنته مع مجموعة تطوير البرمجيات مع الدوت نت.(على نظامي ،يمكن إيجاده من ابدأ Start >> جميع البرامج[All] Programs >> Microsoft Windows SDK v6.0A >>Tools >> Windows Resource Localization Editor.واسم سطر أمره هو *winres.exe*).عندما تكون جاهز على تحويل مترجم الفورم إلى لغة معينة،تحتاج إلى توفيرها لهذا البرنامج،وملف الفورم*.resx* (مثل *Form1.resx*).يحاكي البرنامج عرض الفورم كما تظهر في الفيجوال أستوديو،وتسمح للمترجم بتعديل أي خاصيات فورم أو أداة مناسبة إلى لغة معينة.يبين الشكل التالي فورم Form1التطبيق الذي عملناه منذ حين في محرر التخصيص.



يطلب البرنامج اللغة المستهدفة أو لغة الثقافة عندما تحاول حفظ التغيرات.إنه يعمل على إخراج ملف*.resx* للغة المخصصة (مثل العربية *Form1.ar.resx*)،يمكن أن يتم استخدامها في تطبيقك.حالما تحصل على ملفات الموارد من المترجمات،خزنها(الملفات،وليس المترجمات) في دليل موارد المشروع،واعمل على إعادة بناء المشروع لإنتاج المجمعات التابعة الصحيحة.

ترجمة المصادر بشكل يدوي.Manually Compiling Resources

من المحتمل أن تعمل على إنتاج المجمعات التابعة يدوياً من ملفات*resx.* المصدرية دون إعادة بناء كامل المشروع في الفيجوال أستوديو.سيكون عليك استخدام سطر أمر ويندوز Windows command line (*cmd.exe*)، وستحتاج إمكانية الوصول إلى ملف EXE أو DLLللمجمع.وهذا ليس لزرع القلق في قلبك.
خطوات "إنتاجgenerate "و"ترجمةcompile " يمكن أن يتم عملها باستخدام خدمات سطري أمرcommand-line :
resgen.exe و*al.exe*، ألا يبدو ذلك عظيماً؟.
كما مع أدوات سطر أمر الدوت نت الأخرى،فإن هذه الأدوات تحتاج أن يتم تنصيب بيئة سطر الأمر command-line environment المناسبة،لتضمن أن لديك البيئة الصحيحة،تحتاج إلى فتح نسخة دوت نت خاصة لسطر الأمر.إن مجموعة تطوير برمجيات الدوت نتNET SDK. قد تم تنصيبها عندما عملت على تنصيب إطار العمل،لذلك ستكون قادر على إيجاد مدخلة قائمة البدء من أجل هذا من قائمة ابدأ Start<< جميع البرامجPrograms [All] → <<ميكروسوفت فيجوال أستوديو2008 Microsoft Visual Studio 2008<<أدوات الفيجوال أستوديو Visual Studio Tools << محث أمر الفيجوال أستوديو2008 Visual Studio 2008 Command Prompt.

منتج ملف الموارد.Resource File Generation

حالما يكون متاح لديك ملف*resx.*،إما بإنشائه يدوياً أو باستخدام محرر تخصيص مورد ويندوزWindows Resource Localization Editor،فإنك تعمل على إنتاج ملف الموارد *resources*باستخدام*resgen.exe* ،تقبل خدمة سطر أمر مولد المورد(وهي جزء من مجموع أدوات مجموعة تطوير برمجيات الدوت نت) مدخلات ومخرجات اسم ملف كمعاملات نسبية له:

```
resgen.exe Form1.ar.resx Form1.ar.resources
```

بالطبع عليك أولاً تبديل الدليل إلى دليل المشروع كما يلي:

```
cd c:\foreignnames
```

ومن ثم اضغط انتر وذلك إذا كنت قد حفظت البرنامج في هذا المسار المحدد في الأعلى وبنفس الاسم(foreignnames)وإذا لم تعمل ذلك حاول حفظ المشروع في هذا المسار الذي حددته في الأعلى ،بعد أن تنقر انتر سيتغير لديك المسار داخل سطر الأمر إلى المسار التالي:

```
C:\ForeignNames>
```

إذا أصدرت مخرجات اسم ملف،سيعمل *resgen*ببساطة على استبدال الامتداد *resx.* بـ *resources.*
إذا كان لديك مجمعات لغة أجنبية متعددة(من أجل نماذج متعددة)،اعمل على إنتاج ملفات الموارد لكل النماذج.ومن ثم فإنك ستكون جاهز لترجمة المجمع التابع.

ترجمة المجمعات التابعة(أو الثانوية).Compiling Satellite Assemblies

تستخدم الدوت نت*al.exe* ،برنامج رابط المجمعAssembly Linker program،لترجمة جميع تطبيقات الدوت نت إلى ملفات المجمع النهائي.سنستخدم نفس هذا البرنامج لإنتاج المجمعات التابعة.لقد تم تصميم المعاملات النسبية له بواسطة منظمة سرية،لذلك وللحصول عليها سيأخذ بعض العمل.لنلقي نظرة على الأمر أولاً،ومن ثم سأشرحه.
طبعاً قبل إدخال سطر الأمر هذا لا تنسى أن تغير المسار إلى المسار(C:\ForeignNames)

```
al.exe/target:lib /embed:Form1.ar.resources,ForeignNames.Form1.ar.resources /culture:ar /out:ForeignNames.resources.dll
/template:bin\Release\ForeignNames.exe
```

بالطبع عليك وضع هذا الأمر في سطر أمر واحد ولا تنسى الفراغات بين كل أمر:لاحظ الفراغ مثلا بين والذي لم أعمل على تعليمه(التقطيعات الصغيرة).
الخيارات الموفرة إلى *al.exe*هي التي تقوم بكل هذا السحر:

/target:lib

يقول الجزء lib،"اعمل على إخراج ملف من نوع DLL "

/embed

يشير هذا الجزء إلى الملفات المصدرية التي تريد تضمينها في المجمع الذي ستعمل على إخراجه.الجزء الأول المحدد بفاصلة يشير إلى اسم الملف المصدري،بينما يشير الجزء الثاني إلى الاسم الذي سيعرف به هذا المورد في التطبيق.ويجب أن يكون الاسم بالتنسيق:الاسم القاعدي.الاسم الثقافي.المواردresources.*basename.cultureName*، حيث الاسم القاعدي *basename*هو اسم التطبيق(من أجل موارد التطبيق الواسعة) أو اسم الفئة(مكافئة لفضاء أسماءها)من أجل فئة خاصة،مثل Form1.بما أن تطبيقي وفضاء أسماءه الأعلى الافتراضي كلاهما" ForeignNames "،لذلك عملت على تضمينه في اسم المكون. تستطيع إضافة العديد من خيارات/embed كلما كان لديك ملفات موارد أكثر تحتاج أن تضمنها.

/culture

على الرغم من أنك ستعمل في النهاية على وضع المجمع التابع في مجلد محدد من أجل الثقافة الهدف،فإن الفيجوال بيسك لاتثق بك.بالمقابل،فهي تريد تسجيل الثقافة المضمنة في المجمع نفسه.وتعمل ذلك من خيار سطر الأمر هذا.

/out

يحدد هذا الخيار الاسم الذي سيتم إخراجه للملف التابع(أو الثانوي).في الحقيقة تحتاج إلى استخدام الاسم من أجل الملف*application.resources.dll*، حيث *application* هو نفس اسم تطبيقك الذي قبل اللاحقة *exe.* .وإذا لم تعمل هذا(لم تحدد اسم الملف الذي سيتم إخراجه)فلن يعمل،حسناً،ولكن بإمكانك جعله يعمل بضبط ملف تركيب التطبيق *app.config*،ولكن ذلك الملف مخيف نوعاً ما،ولا تريد الذهاب إلى هناك.

/template

هذا هو الخيار الذي يقول"إني اعمل على صنع مجمع ثانوي،والمجمع الرئيسي ذو الصلة هو ×".
لاستخدام المجمع الثانوي،اعمل على إيجاد الدليل الذي يحتوي على المجمع الرئيسي(EXE).اعمل على إنشاء دليل ثانوي هناك،وامنحه اسم اللغة أو المفتاح الثقافي للغة المستخدم لإنشاء ا لمجمع(" ar"في حالتي)ومن ثم ضع المجمع الثانوي الجديد في ذلك الدليل الثانوي.

ميزات تخصيص أخرى. Other Localization Features

التخصيص أكثر من أن يكون مجرد كلمات على الشاشة.توجد أيضاً قضايا لكيفية عرض الوقت،التاريخ،والقيم المالية للمستخدم.الأخبار الجيدة،هو أن كل ميزة ستعمل بشكل آلي إذا ما عممت *globalize* التطبيق بشكل مناسب.بما أن كل برنامج دوت نت يحتفظ "بثقافة واجهة المستخدم"(والتي تعاملنا معها في عينة البرنامج السابق)،فإن لديها"ثقافة عامة" تستخدم لمعالجة قيم الوقت، التاريخ،التمويل،وأشياء أخرى معتمدة على ثقافة مشابهة.إذا كنت تستخدم الطرق الأساسية مثل CDateلاستخراج قيم التاريخ،بدلاً من عمل بحث على التاريخ المدخل من قبل المستخدم الإنسان،إنك تحصل على معالجة تاريخ خاص بثقافة بدون تعب.وأيضاً،إذا استخدمت تنسيقات محددة مسبقاً من أجل الطريقة (وطرق إخراج سلاسل حرفية أخرى)،فإنك ستحصل على التنسيق الخاص بالثقافة الصحيح بدون أي جهد إضافي.دعنا نجرب مثال سريع يعرض العملة باستخدام العملة المحلية.
اعمل على إنشاء تطبيق نماذج ويندوز جديد.سأعمل على إضافة الكود التالي إلى الملف *ApplicationEvents.vb.*

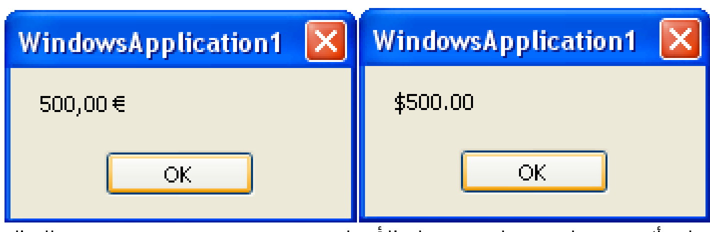
```
Private Sub MyApplication_Startup(ByVal sender As Object, ByVal e As Microsoft.VisualBasic.ApplicationServices.StartupEventArgs) Handles Me.Startup
            If (MsgBox("هل تريد التبديل من العربية إلى الفرنسية؟", MsgBoxStyle.Question Or MsgBoxStyle.YesNo) = MsgBoxResult.Yes) Then
                My.Application.ChangeCulture("fr-FR")
            End If
        End Sub
```

وهذا الكود مطابق للذي استخدمناه سابقاً في المثال السابق،ولكنني أستدعي هنا My.Application.ChangeCulture بدل My.Application.ChangeUICulture (حيث أنني حذفت الجزء UI).وهذا يغير ثقافة معالجة السلاسل الحرفية بدل من ثقافة واجهة المستخدم.

والآن سأعمل على إضافة الكود التالي إلى فئة الفورم.

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        MsgBox(Format("500", "Currency"))
    End Sub
```

يبين الشكل التالي نتائج هذا الكود عند تشغيل التطبيق بين النمطين الإنكليزي والفرنسي:



تتضمن مكتبات فئة إطار العمل ميزات معالجة الثقافة أكثر من ما عرضناه في فضاء الأسماء System.Globalization.تتيح لك الفئات في فضاء الأسماء هذا ضبط مخرجات سلاسل حرفية معتمدة على ثقافة لتلبي حاجاتك.معظمها معد وموجه لمجموعات ثقافية معينة،لذلك فلن أعمل على مناقشتها.

مشروع.

ما سنفعله في مشروع هذا الفصل هو تمكين ما تبقى من ميزات الإدارة الخاصة بالزبون.تلك الميزات التي تتضمن إدارة الغرامات للزبائن السيئي السلوك،والذين لا يعيدون كتب المكتبة في الوقت المحدد.سنستخدم ميزات تنسيق العملة الشاملة التي ناقشنها في هذا الفصل لجعل التطبيق شامل الوصول قدر الإمكان.

تتبع مدفوعات زبون.Tracking Patron Payments

لنعمل على إنشاء فئة تعرض الميزات الهامة لكل مجموعة من المدفوعات المطبقة على بند خاص مدخل.وبالطبع الكل سيتم تخزينه في قاعدة بيانات المكتبة.ولكن الاحتفاظ بملخص للمدفوعات بشكل مؤقت مخزنة في الذاكرة يبسط بعض المعالجة.

أضف فئة بند جديد إلى مشروع المكتبة،وسمها *PaymentItem.vb*، وعرفها باستخدام الكود التالي:

```
Public Class PaymentItem
    ' تستخدم لتتبع وطباعة بطاقات دفع الزبون
    Public ItemTitle As String
    Public PatronCopyID As Integer
    Public FeesPaid As Decimal
    Public BalanceDue As Decimal
End Class
```

كل حالة من هذه الفئة تعرف الغرامات المحسوبة والمدفوعات من أجل بند مكتبة معين(ItemTitle)وللزبون الذي أعاد البند بشكل متأخر(PatronCopyID)

حساب غرامات الزبون.Calculating Patron Fines

نحتاج أيضاً لمعرفة الغرامات الكلية total fines المدين بها زبون من أجل جميع البنود،حتى عندما لا نظهر جميع التفاصيل.أضف الدالة CalculatePatronFinesإلى الوحدة البرمجية *General.vb*.

```
Public Function CalculatePatronFines(ByVal patronID As Integer) As Decimal
        ' تقديم معرف زبون،لحساب الغرامات المستحقة
        Dim sqlText As String

        On Error GoTo ErrorHandler
        ' استخراج سجلات التغرم من أجل الزبون
            sqlText = "SELECT SUM(Fine - Paid) FROM PatronCopy WHERE Patron = " & patronID
        Return DBGetDecimal(ExecuteSQLReturn(sqlText))
ErrorHandler:
        GeneralError("CalculatePatronFines", Err.GetException())
        Return 0@
    End Function
```

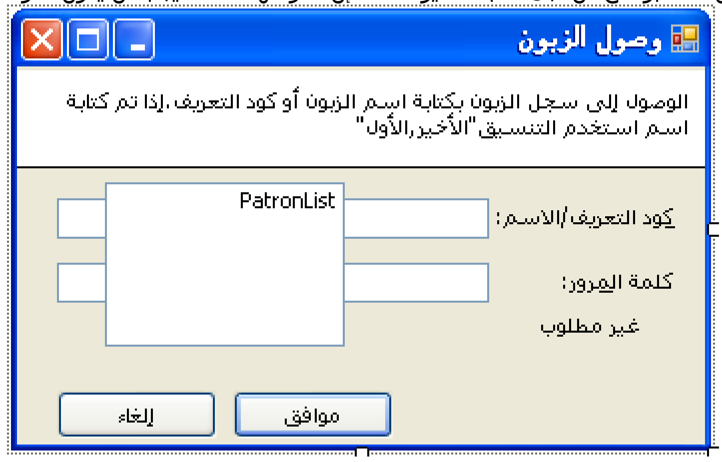
في الحقيقة،بما أن قاعدة البيانات تقوم بجميع العمل من أجل جمع القيم.فعملت على تفحص قاعدة البيانات وأكدت على أن الحقلين Fine وPaid مطلوبين،ولن يكونا"بدون قيمة NULL ".

الوصول إلى سجل الزبون.Patron Record Access

قبل مراجعة سجل الزبون،يجب على المستخدم أن يحدد الزبون.ويتم عمل هذا من خلال نموذج الوصول إلى سجل الزبون،نوعاً ما نموذج دخول للزبون.يتم إسناد كلمة مرور لكل زبون،والتي يجب أن تكون مزودة قبل أن يتمكن الزبون من الوصول إلى سجله.يمكن للمدراء الوصول إلى سجل الزبون دون الحاجة إلى كلمة المرور.لقد عملت على إضافة الفورم

لمشروعك،وهي تظهر في الشكل التالي،وكود هذه الفورم مشابه كثيراً للفورم"تغير المستخدم *PatronAccess.vb* *ChangeUser.vb* "،الفورم التي توفر للمدراء إمكانية الوصول إلى البرنامج،وأضفناها في الفصل11. تتصرف فورم وصول المستخدم بشكل مختلف قليلاً من أجل المدراء والزبائن النظاميين.

.الزبائن النظاميين يجب إما أن يوفروا كود التعريف الخاص بهم،أو أسمهم(الاسم الأخير الاسم الكامل،بشكل اختياري القيمة الشاملة على الاسم الأول) وكلمة المرور الخاصة بهم. إذا كانوا يستخدمون اسم جزئي بدل كود التعريف،والبحث عن ذلك الاسم ينتج عنه تطابقات متعددة،فعليهم أن يوفروا مدخلة أكثر دقةً لأسمائهم.(إذا ما كان زبونين لديهم نفس الاسم، فسيكون عليهم الاعتماد على كود التعريف،ولكن هذا البرنامج من أجل مكتبة صغيرة،لذلك فإن تعارض الأسماء يجب أن يكون نادر الحدوث.



.يعمل المدراء على إدخال اسم الزبون،أو كود التعريف،ولكن ليس هناك حاجة إلى كلمة المرور.

إذا كانت هناك تطابقات متعددة لأسم،فإن الفورم تعمل على جلب جميع الأسماء المتطابقة في القائمة،ويمكن للمدير أن يختار المدخلة الصحيحة من القائمة.وهذا يمنح المدير إمكانية وصول كاملة لجميع سجلات الزبون.من الواضح أهمية تسجيل الخروج للمدير عندما ينتهي من استخدام الجهاز(أو شبكة الأجهزة)التي تكون متاحة للزبائن.

توفر طريقة SelectPatron الفورمPatronAccess واجهة الفورم لكل من المدراء والزبائن العاديين.تعود الدالة بمعرف الزبون ID المختار،أو 1- إذا لم ينجح المستخدم في الوصول إلى سجل الزبون.

تعديل كلمة مرور الزبون.Patron Password Modification

على الرغم من استطاعة المدراء من تغير كلمة مرور كل زبون من خلال "نموذج الزبون *Patron.vb* "،فلا نريد أن نمنح الزبائن العاديين إمكانية الوصول إلى تلك الفورم وكل محتوياتها. ولكن ما زلنا نريد أن يكون الزبائن قادرين على تغير كلمات المرور الخاصة بهم،وبما أنه شيء خاص وسري.عملت على إضافة الفورم *PatronPassword.vb* إلى المشروع لإنجاز هذا الهدف(شاهد الشكل التالي).



هذه الفورم بشكل أساسي مجموعة جزئية منخفضة عن الفورم.بما أنها بحاجة إلى التعامل مع الزبائن الفعالين فقط فليس لديها الكثير من كود النموذج.والذي يميز بين سجلات الزبون الجديدة والموجودة.تركيز نموذج كلمة مرور الزبون على تحديث العبارة التي تضع كلمة مرور الزبون، في الطريقة SaveFormData.

```
Private Function SaveFormData() As Boolean
' يـريـد المستخدم حفظ التغيرات
        Dim sqlText As String
        On Error GoTo ErrorHandler
        Me.Cursor = Windows.Forms.Cursors.WaitCursor
' حفظ البيانات
        sqlText = "UPDATE Patron SET [Password] = " & DBText(EncryptPassword("patron",
Trim(RecordPassword.Text))) & _
            " WHERE ID = " & ActiveID
        ExecuteSQL(sqlText)
' في حال النجاح
        Me.Cursor = Windows.Forms.Cursors.Default
        Return True
ErrorHandler:
        Me.Cursor = Windows.Forms.Cursors.Default
        GeneralError("PatronPassword.SaveFormData", Err.GetException())
        Return False
    End Function
```

الكلمة *Password* كلمة محجوزة في سكول سرفرSQL Server ، لذلك نحتاج إلى تجاوزها من خلال أقواس مضلعة عند الإشارة إلى الحقل في عبارات سكول.

جمع مدفوعات زبون.Collecting Patron Payments

في عالم مثالي،لن يسمح الزبائن أن تبقى كتبهم وبنود المكتبة الأخرى أن تصل إلى حالة فوات الميعاد.بالطبع، في العالم المثالي،سيتيح لك المكتبات الاحتفاظ بالكتب التي تحب بشكل غير محدد.ويريحوني من ملاحظات التأخير التسديد المتواصلة.

ولكن من أجل تلك المكتبات الصغيرة والتي تصر على فرض الغرامات من أجل البنود المتأخرة.فإن مشروع المكتبة يتضمن ميزات من أجل إسناد وتعقب الغرامات.في فصل متأخر،سنعمل على إضافة الكود الذي يحسب الغرامات بشكل تلقائي من أجل البنود المتأخرة.أما الآن،سنعمل على تنفيذ الفورم التي تتيح لك توثيق مدفوعات زبون وتسويات مالية أخرى للبنود في سجل زبون.

عملت على إضافة الفورم PatronPaymentلمجموع ملفات المشروع،ولكنها غير متكاملة حتى الآن مع المشروع.اختر الملف*PatronPayment.vb* في مستكشف الحلول،ومن ثم غير خاصية Build Action (في لوحة الخاصياتProperties) من Noneإلى Compile.يبين الشكل التالي الأدوات على هذه الفورم.

الغرامات التي تضاف بشكل تلقائي على بند متأخر تظهر في حقل قاعدة البياناتPatronCopy.Fine. على الرغم من أن تلك القيمة يتم عرضها على فورم "مدفوعات زبون Patron Payment"،فهي ليست التركيز الرئيسي على تلك الفورم.بالمقابل،تتواجد الفورم للسماح لأمناء المكتبة من إدخال الرسوم والمدفوعات من أجل بند تم إرجاعه مسبقاً، وتخزين هذه التحديثات في جدول قاعدة البياناتPatronPayment. يتعقب هذا الجدول أربع أنواع من أحداث التمويل من أجل كل بند تم إعادته من قبل زبون:

- الغرامات الإضافية المفروضة من قبل أمين مكتبة أو مدير.على سبيل المثال،يمكن أن يضيف قيمة لبند كغرامة إذا ما تم نتج عنه أن الزبون قد فقد (أو أضاع)البند . مدخلات التغريم الإضافية تستخدم الحرف *F*في حقل قاعدة البيانات PatronPayment.EntryType.
- المدفوعات المعمولة من قبل زبون من أجل بند متأخر الإعادة. *P*هو نوع المدخلة.
- عزل (أو صرف)بعض أو جميع الغرامات المعلقة pending fines من أجل بند متأخر.يتم الإشارة إليه بنوع المدخلة *D*.
- إذا كانت نوع المدخلة *R* هو ،فإن السجل يشير إلى مدفوعات مسترجعة refund إلى زبون من قبل المكتبة.



كل سجل جدول PatronPaymentيتضمن تاريخ مداولة transaction، كمية المداولة،تعليقات اختيارية،ومعرف(هوية) المستخدم المدير الذي يسجل المدخلة.لجعل الكود أكثر وضوحاً بقليل،يتم تحويل أكواد الأحرف الأبجدية في جدول قاعدة البيانات إلى قيم عددية من العداد EventEntryType.

```
Private Enum EventEntryType
        NotDefined
        PatronPayment
        FineAdded
        FineDismissal
        RefundToPatron
        OverdueFines
    End Enum
```

تسمح المدخلة OverdueFinesلقيمPatronCopy.Fines لأن تكون جزء من تاريخ التمويل المعروض على الفورم.

يستخدم أمين المكتبة الحقول في مقطع"حدث تسديد جديد" للنموذج PatronPayment لإضافة سجلات المدفوعات والرسوم.جميع السجلات المضافة سابقاً تظهر في القائمة EventHistory،في المقطع"تاريخ حدث التسديد" من الفورم.

الفورم المستدعي(سنضيفه فيما بعد في هذا الفصل)يحتاج أن يمرر القيمة PatronCopy.ID إلى معرف سجل مناسب.ولكن على المشروع أن يحصل على المدفوعات المضافة على هذه الفورم بالعودة إلى الفورم الرئيسية.كلا الفورمين سيتشاركان مجموعة من كائنات PaymentItem باستخدام الفئة التي أضفناها من عدة مقاطع سابقة في هذا الفصل.سنعمل على تخزينها في متغير عضو محلي كمجموعة شاملة.

```
Private PaymentsOnly As Generic.List(Of PaymentItem)
```

نقطة الإدخال في الفورم ستكون طريقة شاملة مسماة ManagePayments.أضف ذلك الكود إلى الفئة PatronPayment.

```
Public Sub ManagePayments(ByVal patronCopyID As Integer, _
              ByVal sessionPayments As Generic.List(Of PaymentItem))
' إدارة المدفوعات من أجل بند معين
        ActivePatronCopyID = patronCopyID
        PaymentsOnly = sessionPayments
```

```
        Me.ShowDialog()
    End Sub
```

تسجل هذه الطريقة رقم لنسخة زبون the patron-copy ID وجمع المدفوعات من أجل بند تم إخراجه.ومن ثم تنتقل المعالجة إلى معالج حدث تحميل Load الفورم.في هذا الروتين والذي سنضيف له كود إدارة التمويل المحلي أو المخصص.في الروتينPatronPayment_Load، يتم عمل بحث حول واحدة إلى ثلاث طرق من خلال طريقة الكود الذي يحمل " التفاصيل الموجزةsummary details" من قاعدة البيانات.بعد السطر التالي تماماً:

```
RecordItem.Text = CStr(dbInfo!Title)
```

أضف العبارات التي بشكل عام تنسق قيم العملة من أجل لصاقات الغرامات،المدفوعات،مختصر الرصيد التي تظهر قرب أعلى الفورم.

```
originalFine = CDec(dbInfo!Fine)
        RecordFine.Text = Format(originalFine, "Currency")
        RecordPayments.Text = Format(CDec(dbInfo!Paid), "Currency")
        balanceDue = originalFine - CDec(dbInfo!Paid)
        RecordBalance.Text = Format(balanceDue, "Currency")
        dbInfo.Close()
```

باقي كود معالج حدث التحميل يحمل السجلات الموجودة من الجدول PatronPayment،زائد التغريم المتأخر الأصلي،لأي،من حقول قاعدة البيانات PatronCopy.Fine.

فيما بعد عندما ينقر المستخدم على الزر "إضافة" لإضافة حدث تمويل جديد إلى مدخلة نسخة بند وزبون،الروتين SaveEventDataمكافئ إلى الطريقة SaveFormData في أغلب النماذج الأخرى التي طورناها حتى الآن،تحفظ المعلومات المحدثة في قاعدة البيانات.يحتاج هذا الروتين إلى حفظ الرسوم الجديدة أو المدفوعات في الجدول PatronPayment، زائد تحديث ملخص المدفوعات والرسوم في السجلPatronCopy . أضف الكود الذي يكتب هذه السجلات،بعد الحسابات من أجل المتغيرات fineAmount و paidAmount،في الطريقة SaveEventData.

```
'إضافة المدخلة إلى قاعدة البيانات
        TransactionBegin()
        sqlText = "INSERT INTO PatronPayment (PatronCopy, EntryDate, EntryType, " & _
            "Amount, Comment, UserID) OUTPUT INSERTED.ID VALUES (" & _
            ActivePatronCopyID & ", GETDATE(), " & DBText(entryCode) & ", " & _
            RecordAmount.Text & ", " & DBText(Trim(RecordComment.Text)) & _
            ", " & LoggedInUserID & ")"
        newID = CInt(ExecuteSQLReturn(sqlText))
        sqlText = "UPDATE PatronCopy SET Fine = " & fineAmount & _
            ", Paid = " & paidAmount & " WHERE ID = " & ActivePatronCopyID
        ExecuteSQL(sqlText)
        TransactionCommit()
```

عملت على إنهاء كل من عبارات قاعدة البيانات في المداولة للمساعدة على ضمان تكامل البيانات.حالما يتم تحديث قاعدة البيانات،عندها يحين وقت تحديث الشاشة(أو العرض)، القائمة التي على الشاشة للرسوم والمدفوعات تحتاج هذا السجل الجديد.تستخدم تلك القائمة الفئة EventHistoryItem،تشكيلة عن فئة التطبيق الواسعة ListItemDataالتي نستخدمها عادة في أداة صندوق القائمةListBox. لـ EventHistoryItemحقول والتي هي خاصة بمعلومات التمويل المعروضة في صندوق القائمةEventHistory. أضف الكود الذي يبني سجل EventHistoryItemويضيفه إلى القائمةEventHistory، مباشرةً بعد كود تحديث قاعدة البيانات الذي أضفناه منذ قليل.

```
'أضف بند إلى قائمة الإدخال
        historyItem = New EventHistoryItem
        historyItem.PaymentID = newID
        historyItem.EntryDate = Today
        historyItem.PaymentAmount = CDec(RecordAmount.Text)
        historyItem.Comments = Trim(RecordComment.Text)
        historyItem.EntryType = entryType
        EventHistory.Items.Add(historyItem)
```

هذا الكود متبوع بكود مشابه يحدث القائمةPaymentsOnly، الـ Generic.List(Of PaymentItem)التي تم تمريرها من الفورم المستدعي.إما يحدث الكود سجل المدفوعات الموجودة،أو يضيف سجل جديد إلى القائمة الشاملة.

```
'أضف مدفوعات جديدة
            scanPayment = New PaymentItem
            scanPayment.PatronCopyID = ActivePatronCopyID
            scanPayment.ItemTitle = RecordItem.Text
            scanPayment.FeesPaid = paidAmount
            scanPayment.BalanceDue = fineAmount - paidAmount
            PaymentsOnly.Add(scanPayment)
```

قبل مغادرة الدالة،نحتاج إلى تنشيط قيم ملخص التمويل الثلاث قرب أعلى الفورم،التي نعمل على وضعها عندما يتم تحميل الفورم للمرة الأولى.أضف هذا الكود بعد تحديث القائمة PaymentOnly.

```
'تحديث القيم التي على الشاشة
        RecordFine.Text = Format(fineAmount, "Currency")
        RecordPayments.Text = Format(paidAmount, "Currency")
        RecordBalance.Text = Format(fineAmount - paidAmount, "Currency")
```

القائمة EventHistoryهي أداة حامل رسم لسطر متغير الارتفاع،مشابهة لما صممناه في الفصل 18،يعمل حدثها MeasureItemعلى وضع ارتفاع كل بند قائمة(تظهر التعليقات على السطر الثاني عندما يكون متاح)،ومعالج حدثها DrawItemيعمل الرسم الفعلي لكل عمود بيانات ولكل التعليقات.

إدارة جميع المدفوعات والغرامات.Managing All Fines and Payments

تتيح فورم مدفوعات زبون Patron Payment لأمين المكتبة من إدخال غرامات ومدفوعات مستقلة،ولكن مايزال البرنامج بحاجة إلى فورم تدير جميع المدفوعات والغرامات من أجل زبون واحد، الفورم التي تعمل على إظهار فورم مدفوعات زبون عند الحاجة.تنجز الفورم"سجل زبون*PatronRecord.vb* " هذه الحاجة.عملت على إضافة هذه الفورم

إلى المشروع،على الرغم من أنك بحاجة إلى تمكينها،كما مر من قبل اعمل على تمكينها كما مكنا الفورم السابقة(بشكل عام هذا للتتعلم كيف تعطل وتمكن فورم فإني عملت على تمكين كل النماذج في المشروع).

وهذه الفورم متاحة لكل من المدراء والزبائن،على الرغم من أن بعض الحقول مخفية عن الزبون.

يقود زر"كلمة المرور Password "إلى الفورم"تغير كلمة مرور الزبونChange Patron Password " التي أضفناها سابقاً في هذا الفصل.الزر"تحرير Edit "،متاح فقط للمدراء،يوفر إمكانية الوصول إلى كامل النموذج*Patron.vb* . يعرض المقطع الرئيسي لسجل الزبون قائمة بجميع بنود الزبون التي تم إخراجها.وهي تتضمن الزر"تجديد Renew" والذي يتيح للزبون تمديد تاريخ الإعادة من أجل بند تم إخراجه.سنعمل على إضافة الكود من أجل تلك الميزة في فصل لاحق.

تعرض الفورم أيضاً ملخص لجميع الغرامات المعلقة والمدفوعات.يبين الشكل التالي تبويب الغرامات وحقوله.



يعمل الزر "طباعة بطاقة الرصيد" على إنتاج وصل مطبوع لجميع الغرامات والمدفوعات من أجل زبون ما.وسنعمل على إضافة كوده في فصل لاحق.

معظم الكود في هذه الفورم موجود لإدارة الغرامات والمدفوعات.لإضافة رسم أو تسديد،يختار أمين المكتبة بند من قائمة الغرامات،ومن ثم ينقر الزر"الغرامات والمدفوعات". وهذا يحضر الفورم"مدفوعات زبون "التي أضفناها للتو.القائمتين الرئيسيتين على فورم "سجل الزبون" لا يستخدمان الفئة القياسيةListItemData ، ويستخدمان فئة مناسبة أكثر لدعم العرض الضروري لكل قائمة.وسنعمل على إضافة هذه الفئة PatronDetailItemكفئة عامة بما أنها(كما سنرى في الفصل القادم)ستستخدم في مكان أخر أيضاً في مشروع المكتبة.اعمل على إنشاء فئة جديدة سمهاPatronDetailItem.vb ، واستخدم الكود التالي من أجل محتواها.

```
Public DetailID As Integer
    Public TitleText As String
    Public DueDate As Date
    Public FineAmount As Decimal
    Public PaidAmount As Decimal
    Public BalanceDue As Decimal
```

والآن ارجع إلى الفورم"سجل زبون".كما ترى تعرض قائمة "الغرامات" عدة أعمدة لقيم العملة.لنعمل على إضافة الكود والذي ينسق بشكل صحيح العملة تبعاً لإعدادات العملة المحلية(أو الإقليمية).أولاً،ابحث عن الطريقةRefreshPaymentFines. يعمل هذا الروتين على إضافة جميع الغرامات والمدفوعات ويعرض النتيجة من خلال أداة اللصاقة "الرصيد الباقي المستحق".

أعلى هذا الروتين تعليق ينص على "نظف القائمة الحالية".أضف الكود التالي بعد هذا التعليق:

```
نظف القائمة الحالية
        Fines.Items.Clear()
        totalBalance = 0@
        BalanceDue.Text = Format(0@, "Currency")
        Me.Cursor = Windows.Forms.Cursors.WaitCursor
```

يمكن أن تكون لدينا مجموعة حقل"الرصيد الباقي المستحق" كـ"$0.00" "ولكن لن يكون هذا معمم بشكل مناسب.باستخدام الدالة "Format "مع " Currency " كقاعدة تنسيق ولكن تبقى تعرض النتائج بالتنسيق " $0.00"كما ترى بما أنني استخدم على جهازي كواجهة اللغة الإنكليزية والعملة المخصصة هي الدولار،ولكنها تضبط بشكل مناسب التنسيق من أجل الثقافات الأخرى.

تعمل الطريقة RefreshPaymentFines كل تجزيء الحسابات،وتنتهي بالرصيد المتبقي المستحق على زبون في المتغير المحليtotalBalance . .اعمل على إيجاد التعليق الذي ينص على " إظهار الرصيد الكلي"،وأضف الكود التالي بعد هذا التعليق تماماً.

```
' إظهار الرصيد الكلي
        BalanceDue.Text = Format(totalBalance, "Currency")
        ActFinePay.Enabled = False
```

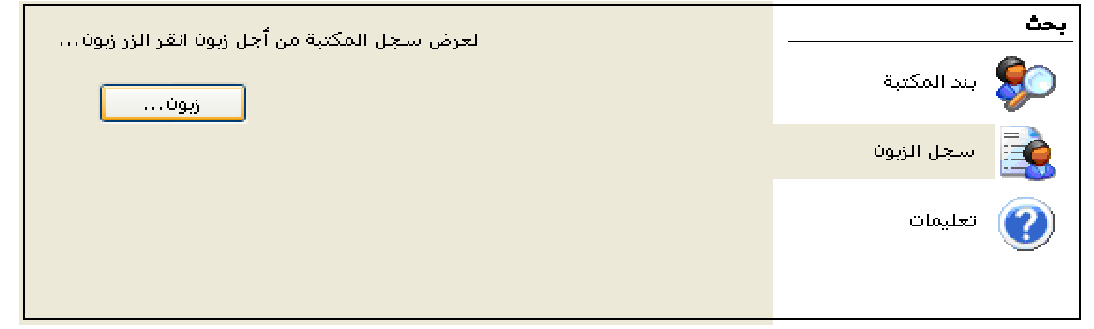
في قائمة الغرامات،تنفيذ حامل رسم صندوق القائمةListBox، يعرض أيضاً قيم العملة.وتستخدم القائمة الأخرى الفئة PatronDetailItemبدل القائمة القياسية ListItemDataمن أجل إدارة بنودها.اعمل على إيجاد الطريقةFines_DrawItem، والتعليق"استخراج التفاصيل من بند قائمة" ضمن هذه الطريقة.وأضف الكود التالي بعد التعليق.

```
' استخراج التفاصيل من بند قائمة
        itemDetail = CType(Fines.Items(e.Index), PatronDetailItem)
        titleText = itemDetail.TitleText
        fineText = Format(itemDetail.FineAmount, "Currency")
        paidText = Format(itemDetail.PaidAmount, "Currency")
        balanceText = Format(itemDetail.BalanceDue, "Currency")
        If (itemDetail.BalanceDue = 0@) Then useNotice = useBrush
```

ينسق هذا المقطع كل قيمة عملة،بشكل افتراضي،كل القيمة العائدة تظهر باللون الأحمر في القائمة.السطر الأخير في هذا المقطع من الكود يجدد اللون إلى لون بند القائمة الحيادي(العادي) إذا لم يكون هناك رصيد عائد.

Connecting Patron Features to the Main Form.ربط ميزات زبون إلى الفورم الرئيسية

بقي علينا تمكين إمكانية الوصول إلى النماذج الخاصة بزبون من خلال فورم المكتبة الرئيسي.كل ما نحتاج عمله هو إضافة معالج حدث من أجل الزر"زبون"



حدد معالج حدث النقر ActAccessPatron_Click على هذا الزر في الكود المصدري لهذه الفورم.ومن ثم أضف الكود التالي لمعالج الحدث هذا.

```
Private Sub ActAccessPatron_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
ActAccessPatron.Click
البحث عن سجل الزبون الفعال                    '
        Dim patronID As Integer
تقديم معرف الزبون            '
        patronID = (New PatronAccess).SelectPatron()
        If (patronID = -1) Then Return
إظهار سجل الزبون             '
        Call (New PatronRecord).ViewPatronRecord(patronID, True)
    End Sub
```

يعمل الكود استدعاء مباشر إلى كلا النموذجين اللتين عملنا على إضافتهما في هذا الفصل:"وصول الزبون" و"سجل زبون".في البداية تظهر الفورم "وصول الزبون" لاختيار سجل الزبون،ومن ثم تعرض تفاصيله من خلال الفورم"سجل زبون".

Dueling Patron Management Forms.التنافس بين نماذج إدارة الزبون

لنعمل تغير أخر بخصوص سجلات الزبون.عملنا على تضمين الزر"إدارة بنود زبون" على فورم "الزبون*Patron.vb*". ويوجد هذا الزر لتوفير إمكانية الوصول إلى الفورم المستقبلي "سجل زبون*PatronRecord.vb*"، ولكنها ما تزال حمل ثقيل حتى الآن.ولكن مع فورم"سجل الزبون *PatronRecord.vb* " الجاهزة،فنحن جاهزين لصنع تاريخ إدارة زبون.

افتح الكود المصدري للنموذج"الزبون*Patron.vb*" واعمل على إيجاد معالج الحدث " ActItems_Click ".ومن ثم أضف الكود التالي له.

```
Call (New PatronRecord).ViewPatronRecord(ActiveID, False)
```

هذا كل شيء،ولكن ومن المحتمل أنك تفكر بنفسك،"تسمح لك نموذج الزبون الآن فتح نموذج سجل زبون.وتلك الفورم لديها زر "تحرير"يتيح لك مرةً أخرى فتح فورم الزبون. وإذا ما كان أمين المكتبة وغد،فهناك الملايين من نماذج إدارة الزبون على الشاشة في الحال.)وهذا صحيح تماماً،لذلك كان علينا إضافة بعض الكود لمنع ذلك من الحدوث.المعامل النسبي الثاني"خطأ" بالنسبة للعلامة PatronRecord.ViewPatronRecordتقول"لاتظهر الزر "تحرير"على فورم"سجل الزبون".وبشكل مشابه يوجد كود في فورم"سجل زبون" تعمل على إيقاف التكرار المستمر.

```
عرض نموذج الزبون'
        If ((New Patron).EditRecordLimited(ActivePatronID) <> -1) Then
```

تخفي الطريقة EditRecordLimitedزر "إدارة بنود زبون" على فورم "الزبون*Patron.vb*". مهما تكن الفورم التي تبدأ بها،تستطيع الوصول إلى الفورم الأخرى،ولكن لن تكون قادر على إنتاج نسخة جديدة من الفورم الأولية.كان بإمكاننا عمل تغيرات ثقافية أكثر.على سبيل المثال،عمود"تاريخ الإعادة" في قائمة"البنود المخرجة" على الفورم"سجل زبون *PatronRecord.vb* " تستخدم تنسيق بيانات ثابتة(لايمكن تغيرها).

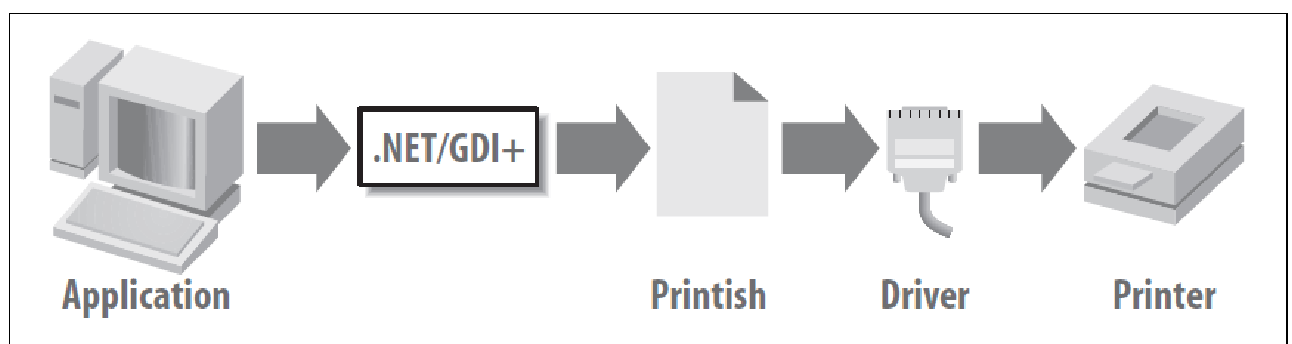
```
dueDate = Format(itemDetail.DueDate, "MMM d, yyyy")
```

كان بإمكانك تغير هذا التاريخ القصير أو أي إعداد ثقافي محايد آخر.

الطباعة Printing

الطباعة في ويندوز Printing in Windows

ينفذ ويندوز نظام من المشغلات (التعريفات) الخاصة بالطابعات printer-specific drivers والتي تحمي المبرمج من معظم تنوعات الطابعات. و هذه المشغلات تتكلم جميعها لغة مشتركة _لندعوها"الطبعة Printish " والتي يترجمها المشغل ضمن اللسان الأصلي للطابعة.
وكمبرمجين،نحتاج فقط إلى تصميم البرمجيات التي تتكلم "الطبعة Printish ".يعمل نظام الطباعة في إطار عمل الدوت نت على إضافة مستوى أخر من ترجمة اللغة.حيث أن برامج دوت نت لا تتصل مباشرةً مع مشغلات الطابعة.بدلاً عنه،تستخدم GDI والأوامر commands ــ نفس الأوامر المستخدمة لتحديثات الشاشة ــ لاستخراج محتوى إلى رسومات طابعة في الذاكرة.يعمل إطار العمل بعدها على تحويل هذه الأوامر إلى طبعة ويرسل المخرجات إلى مشغل الطابعة المناسبة.وأخيرا إلى الطابعة.يبين الشكل التالي ملخص للخطوات المستخدمة في الطباعة في الدوت نت.



الطباعة في الدوت نت Printing in .NET

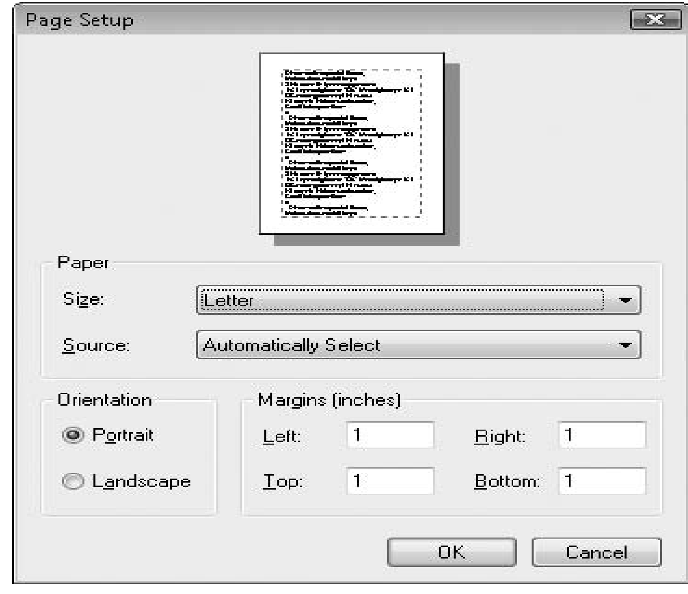
للشاشة والطابعة مخرجات ناتجة من خلال نفس GDI والأوامر بحيث من الممكن أن أجعل هذا الفصل قصير جداً،مع الإشارة إلى الفصل 18 للمزيد من التفاصيل.ولكن هذا يعني أيضاً أن هناك حاجة لأن تكون لوحة – كائن System.Drawing.Graphics - حيث أن الطابعة أو (GDI+ commands) تستهدف مخرجات لهذا الكائن.توفر لك الفئة System.Drawing.Printing.PrintDocumentلوحة مخرجات تحتاجها لكل من مخرجات الطباعة العادية ومعاينة قبل الطباعة.
توجد ثلاث طرق لاستخدام فئة "طباعة مستند PrintDocument " :

- أضف أداة "طباعة مستند PrintDocument " إلى نموذج من صندوق أدوات نماذج ويندوز.تظهر هذه الأداة بشكل افتراضي في مقطع الطباعة لصندوق الأدوات.اسند لها خاصيات ومجيبات أحداث كما مع أي أداة أخرى.
- أنشئ متغير(حقل)على مستوى حالة(نسخة) من فئة "طباعة مستند PrintDocument ".ضمن عبارة في التعريف لتحصل على هيئة الحدث.
- أنشئ حالة(نسخة)محلية من "طباعة مستندPrintDocument " واربط أي حدث باستخدام إضافة معالج AddHandler.

إن هذه طرق قياسية في الدوت نت،ولكن التنوع الذي تملكه أداة ما يجعل الفئة أكثر ملائمة.سندخل ضمن كود طباعة فعلي بعد قليل.
يوجد أربع أدوات خاصة بالطابعة متاحة لمشاريع نماذج ويندوز:

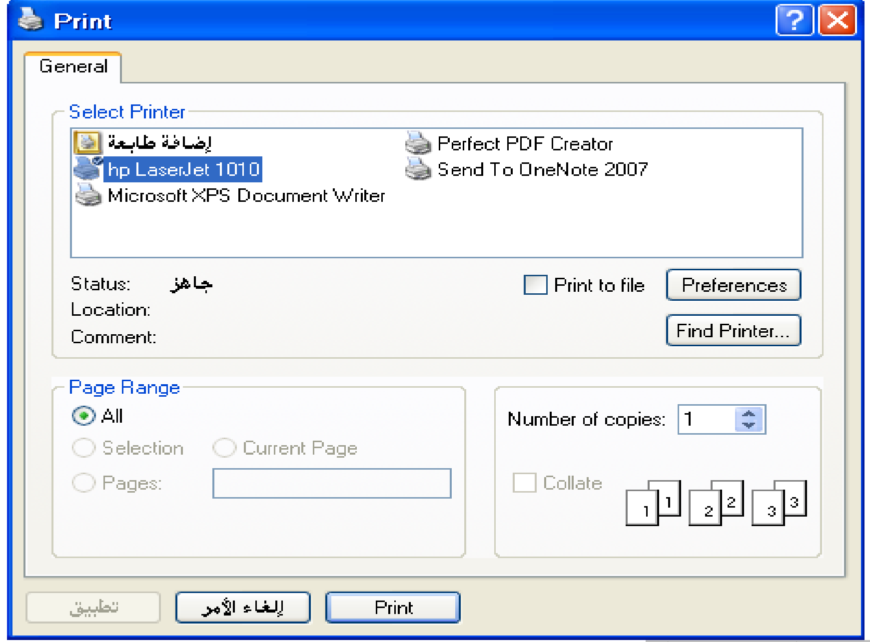
"حوار إعداد الصفحة PageSetupDialog "

تقدم هذه الأداة حوار إعداد طابعة ويندوز قياسي توفر للمستخدم إعداد عمل طباعة خاصة،أو جميع أعمال الطباعة للتطبيق.تعرض الطريقة "إظهار الحوار ShowDialog "للأداة نموذج مبين في الشكل التالي.تعرض الأداة أيضاً خاصيات مرتبطة باختيار المستخدم.تعرض أعضاءها " PageSettings" أولويات للمستخدم خاصة تم تحديدها على الفورم(النموذج).تحدد أعضاء " PrinterSettings"الطابعة المختارة وخاصياتها.
بإمكانك مراجعة هذه الأعضاء وفيما بعد أسندها إلى الفئات الخاصة بالطابعة الأخرى والتي تتضمن نفس الأعضاء.



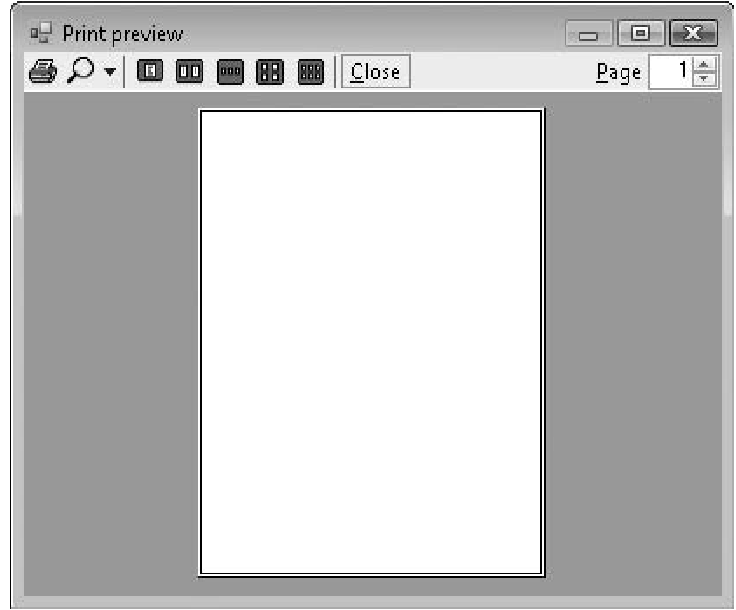
"حوار طباعة PrintDialog"

يظهر الشكل التالي حوار هذه الأداة،الحوار القياسي الذي يظهر في معظم البرامج عندما يختار المستخدم ملف File>> Print أمر قائمة طباعة. وتعرض هذه الأداة أيضاً أعضاء"إعدادات الطابعة PrinterSettings " المستخدمة لإسناد أو استخراج الطابعة المختارة والخيارات المرتبطة بها.



"PrintPreviewDialog حوار معاينة قبل الطباعة"

يعرض حوار هذه الأداة معاينة قبل الطباعة لمستندك المطبوع للمستخدم.وتتضمن مواصفات تقييدية معاينة قياسية من ضمنها قيمة العدسة zoom level وحالة التحكم بإظهار الصفحات.ويعمل زر الطباعة Print button الذي ضمنها على إرسال محتوى المعاينة إلى الطابعة الافتراضية(دون السؤال عن الطابعة المختارة).تتفاعل هذه الأداة مباشرةً مع نسخة "مستند الطباعة PrintDocument " الخاص بك،والذي يقود محتوى العرض الفعلي، يبين الشكل التالي حوار معاينة الطباعة،على الرغم من عدم وجود محتوى خاص بصفحة.



تحتوي أداة "حوار معاينة قبل الطباعة PrintPreviewDialog " مواصفات إدارة معاينة قبل الطباعة الأساسية(مثل ميزة العدسة)والتي تلبي حاجات معظم التطبيقات. لسوء الحظ،إن الأداة محاطة بصندوق أسود،ولاتستطيع بسهولة إضافة مواصفات خاصة بك لها أو إزالة المواصفات التي لاتريدها. توفر الأداة" PrintPreviewControl"واجهة بديلة والتي تسمح لك من أن تخصص بالكامل اختبار معاينة الطباعة.بدلاً من صندوق الحوار الكامل المبين في الشكل السابق،فإن هذه الواجهة تعالج فقط القسم الذي يعرض الصفحة فقط من الفورم.يجب عليك أن تعالج جميع الأشرطة الأدوات والميزات الأخرى،وتربط تخصصها الوظيفي مع أداة المعاينة قبل الطباعة.لن اشرح هذه الأداة في هذا الفصل .
قبل أن تطبع،تحتاج لمعرفة أي طابعة يريد أن يستخدمها مستخدمك لطباعة المخرجات،ومن المحتمل أن تحتاج لمعرفة المواصفات المتاحة للطابعة،مثل دعم الألوان،فإذا كنت مبرمج فيجوال بيسك6،فقد كنت معتاد على مجمع طابعات مريحconvenient Printers collection. إن غياب absenceذلك المجمع في فيجوال بيسك2008 يعني أنه يجب علينا وسائل أكثر غير مباشرة للوصول إلى الطابعات.
تتضمن الفئة System.Drawing.Printing.PrinterSettings مجمع متغير نصي متشارك للطابعات التي تم تثبيتها والذي يجدول مسار كل طابعة مركبة.بإمكانك إسناد أي من هذه النصوص إلى عضو"اسم الطابعة PrinterName " "لإعدادات الطابعة PrinterSettings'"،لجعل طابعة ما متاحة ضمن التطبيق .مقطع الكود التالي يسمح للمستخدم من اختيار من قائمة الطابعات ،ويعرض بعض المعلومات الأساسية حول الطابعة المختارة.

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        ' ----- Display information about the selected printer.
        Dim selectedPrinter As Drawing.Printing.PrinterSettings
        If (ListBox1.SelectedIndex = -1) Then Return
        selectedPrinter = New Drawing.Printing.PrinterSettings()
        selectedPrinter.PrinterName = ListBox1.Text
        MsgBox(selectedPrinter.ToString)
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        ' ----- Display the list of printers.
        Dim scanPrinter As String
        For Each scanPrinter In Drawing.Printing. _
        PrinterSettings.InstalledPrinters
            ListBox1.Items.Add(scanPrinter)
        Next scanPrinter
    End Sub
End Class
```

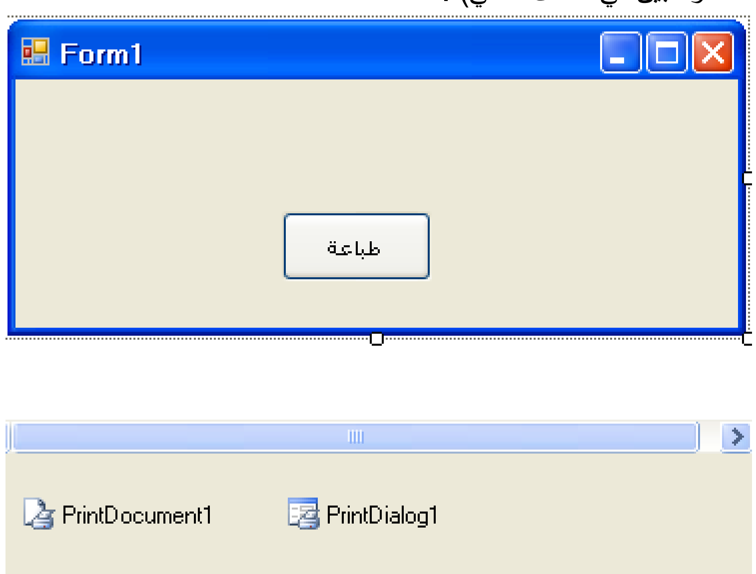
طباعة مستند ما Printing a Document

رأيت مسبقاً أن العديد من مكونات ويندوز تعمل مع بعضها لتوليد مخرجات الطباعة.ضمن كودك للدوت نت،ستستخدم أيضاً العديد من المكونات(الفئات) لقيادة معالجة الطباعة.يتم تضمين أربع خطوات رئيسية(بشكل مباشر على الأقل) في طباعة مستند من ضمن كودك:

1. إنشاء نسخة(حالة) من فئة "طباعة مستند PrintDocument "(أو إضافتها كأداة لنموذجك).
2. وضع إعدادات الطابعات المتنوعة لـ"طباعة مستند PrintDocument's"،إما باستخدام "حوار الطباعة PrintDialog "(أو ما يتعلق بـ)الأداة/الفئة،أو باستخدام الافتراضي أو عمل إعدادات يدوية.

3. إضافة معالج حدث "أطبع صفحة PrintPage " لـ"مستند الطباعة PrintDocument's ." هذا الحدث يتم استدعاءه مرة لكل صفحة،ويستقبل كائن من أجل لوحة الطباعة printer canvas،يطبع كود معالج حدثك صفحة مفردة،ويحدث updates إشارة تخبر فيما إذا يوجد صفحات أخرى أكثر للطباعة.
4. استدعاء الطريقة "طباعة Print " لـ"طباعة مستند PrintDocument's " لبدء عملية الطباعة والدوران.

لنجرب كود صغير لرؤية كيف تبدو عملية الطباعة.وكيف سيبدو الحال فيما يخص برنامج يطبع مستند من خمس صفحات على طابعة المستخدم المختارة؟ستكون المخرجات عدد كبير من الصفحات الرقمية المفردة large single-digit page number، أولاً،لنعمل على إنشاء تطبيق ويندوز جديد new Windows Forms application، ونعمل على إضافة زر وحيد buttonإلى Form1ولنمنحه اسم ActPrint، وسنضيف أيضاً أداة PrintDocumentونسمها(CountingDoc)وأداة PrintDialogونسمها(UserPrinter) يبين الشكل التالي النموذج والأدوات المزود بها(بإمكانك ترك الأسماء الافتراضية لهذه الأدوات كما هو مبين في الشكل التالي) .



تنفذ هذه الأدوات أول خطوتين من الخطوات الأربع لمعالجة الطباعة.وفيما يلي سوف نعمل على إضافة الكود المصدري.

```
Public Class Form1
    Private WhichPage As Integer
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        ' ----- Prompt the user for printer settings, and
        ' start printing.
        PrintDialog1.Document = PrintDocument1
        If (PrintDialog1.ShowDialog() = _
        Windows.Forms.DialogResult.OK) Then _
        PrintDocument1.Print()
    End Sub
    Private Sub CountingDoc_BeginPrint(ByVal sender As Object, _
ByVal e As System.Drawing.Printing.PrintEventArgs)Handles PrintDocument1.BeginPrint
        ' ----- Start the counting over.
        WhichPage = 1
    End Sub
    Private Sub CountingDoc_PrintPage(ByVal sender As Object, _
ByVal e As System.Drawing.Printing. _
PrintPageEventArgs) Handles PrintDocument1.PrintPage
        ' ----- Print a single page.
        Dim hugeFont As Font
        Dim centeredText As StringFormat
        ' ----- Let's go overboard on the font: 256 points!
        hugeFont = New Font("Arial", 256)
        ' ----- Center the text on the page.
        centeredText = New StringFormat()
        centeredText.Alignment = StringAlignment.Center
        centeredText.LineAlignment = StringAlignment.Center
        ' ----- Print the number.
        e.Graphics.DrawString(CStr(WhichPage), hugeFont, _
        Brushes.Black, e.MarginBounds, centeredText)
        ' ----- Draw the page margins to make it clear where
        ' they are.
        e.Graphics.DrawRectangle(Pens.Blue, e.MarginBounds)
```

```
        ' ----- Limit the output to five pages.
        WhichPage += 1
        If (WhichPage <= 5) Then e.HasMorePages = True _
        Else e.HasMorePages = False
    End Sub
End Class
```

ينفذ هذا الكود الخطوة الثالثة(`Button1_Click`) والخطوة الرابعة (`CountingDoc_PrintPage`).يربط معالج حدث نقر الزر `Button1.Click` المستند document وحوار الطباعة Print dialogلذلك فإن كلاهما يشيران إلى نفس الإعدادات.ومن ثم يطلب معالج الحدث هذا من المستخدم اختيار طابعة والخيارات العديدة من خلال استدعاء "إظهار الحوار ShowDialog ".فإذا نقر المستخدم الزر موافق OK على صندوق الحوار هذا،فإنه يعمل على إطلاق استدعاء لطريقة "طباعة Print" الخاصة بالمستندdocument's .

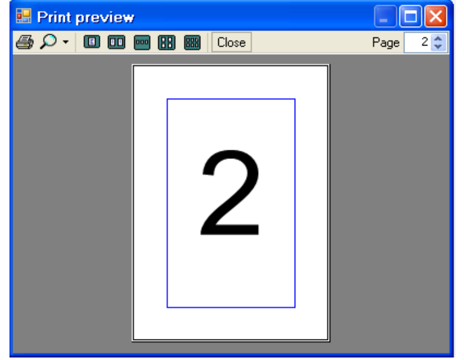
من ثم ينتقل الإجراء إلى أحداث نسخة " PrintDocument".كما ترى فقد قمت بمعالجة أثنين من أحداث PrintDocument:معالج حدث "بدء الطباعة BeginPrint " والذي ينجز بعض العمليات التمهيدية للطباعة،ومعالج حدث "اطبع صفحة PrintPage " والذي يقوم بالعمل الحقيقي .(يوجد أحداث أخرى مثل "أنهي الطباعةEndPrint "،والذي يستخدم للتنظيف عند اكتمال الطباعة،و"استعلام إعدادات الصفحة QueryPageSettings " والذي بإمكانك استخدامه لتغير اتجاه وإعدادات كل صفحة من المستند).عملياً،هذا ليس بالعمل الصعب وخاصة إن كنت على دراية بالغرافيك،والاختلاف الكبير يكمن في الكم من المساحة المتاحة على صفحة الطباعة،والتي تسمح لنا بالتلاعب بالخطوط في المئات من النقاط لحجم الخط.

يبين الشكل التالي الصفحة2 من المخرجات الناتجة عن هذا البرنامج .تستطيع أن ترى في الزاوية اليسارية الدنيا أنها تسجل بشكل كامل مخرجات خمس صفحات.

معاينة قبل الطباعةPrint Preview

إن إضافة واجهة معاينة قبل الطباعة سهل جداً،دعنا لكي نجعل التطبيق يعرض معاينة قبل الطباعة،أضف زر Buttonجديد، وسنضيف أيضاً أداة "حوار معاينة الطباعة PrintPreviewDialog ".حالما تعمل على إضافة ووضع الأدوات السابقة في مكانها ،أضف معالج حدث نقر الزر الثاني التالي:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
        ' ----- Display a preview of the document.
        PrintPreviewDialog1.Document = PrintDocument1
        PrintPreviewDialog1.ShowDialog()
    End Sub
```



كما ترى هذا الكود بسيط جداً.

حساب وترقيم الصفحات Counting and Numbering Pages

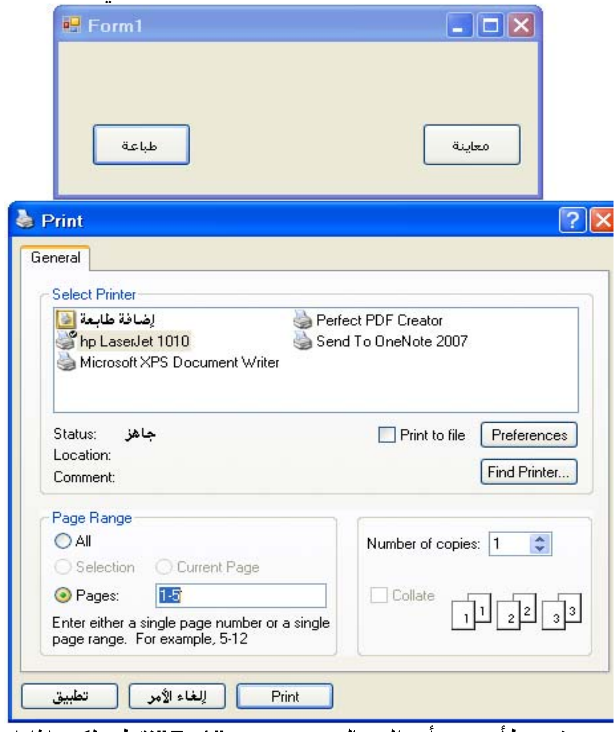
خلال عملية الطباعة(أو المعاينة)،يتم استدعاء معالج حدث "أطبع صفحة PrintPage " لأداة PrintDocument'sمرة لكل صفحة مخرجات. ولكن يوجد شيء يتطلب براعة:فعندما تم استدعاء معالج الحدث "أطبع صفحة PrintPage " للمرة الأولى لم يكن لطباعة "أول صفحة" من المستند،ولكن لطباعة "الصفحة الأولى في حاجة الطباعة"، ومهما يكن رقم الصفحة.ابحث في جميع خاصيات فئة(أو أداة)"طباعة مستند PrintDocument"، ولكن لن تجد أبداً خاصية"عدد الصفحة PageNumber "،ففئة "طباعة مستند PrintDocument " لا تعلم أي شيء حول عدد الصفحات في مستندك،وعلى الرغم من كل شيء فهذا لايهم.كل ما تعلمه أن هناك حزمة من الصفحات بحاجة طباعة،وستعمل على استدعاء معالج حدث "أطبع صفحة PrintPage " حتى تقول "كافيenough ."

إذا رجعت إلى الشكل الثالث،سترى أن حوار الطباعة يتضمن مقطع "مجال الصفحاتPage Range " على الرغم من أن جميع أدواته غير فعالة بشكل افتراضي.فإن أداة "حوار الطباعة PrintDialog " تتضمن ثلاث خاصيات منطقية Booleanوالتي تتيح لك تمكين أدوات خاصة في ذلك المقطع:"السماح للصفحة الحاليةAllowCurrentPage"، "السماح لبعض الصفحاتAllowSomePages "،و"السماح لما تم اختياره AllowSelection ". فوضع أي من هذه الخاصيات إلى صح Trueيمكن خيار الأداة الموافقة.فيما بعد وبعد أن يختار المستخدم،بإمكانك الاستعلام عن خاصية " PrinterSettings.PrintRange " لكائن"طباعة مستند PrintDocument " لتحديد أي خيار هو.

لنعمل على إضافة كود يمكن اختيار مجال الصفحات.وسنبقي على محدودية الصفحات المسموحة فقط للصفحات المرقمة لخمسة فقط،ولكن بإمكان المستخدم أن يكون قادر على اختيار مجال فرعي ضمن المجموعة.ارجع إلى معالج حدث النقر للزر الأول،وأدخل عدة أسطر من الكود وهي بالخط الغامق:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        ' ----- Prompt the user for printer settings, and
        ' start printing.
        PrintDialog1.Document = PrintDocument1
        PrintDialog1.AllowSomePages = True
        PrintDocument1.PrinterSettings.MinimumPage = 1
        PrintDocument1.PrinterSettings.MaximumPage = 5
        PrintDocument1.PrinterSettings.FromPage = 1
        PrintDocument1.PrinterSettings.ToPage = 5
        If (PrintDialog1.ShowDialog() = _
        Windows.Forms.DialogResult.OK) Then _
        PrintDocument1.Print()
    End Sub
```

فعندما ينقر المستخدم على زر "طباعة" وهو نفس الزر الأول والذي سميته طباعة أما الزر الثاني فقد سميته معاينة ،فسترى أن مقطع مجال الصفحات لصندوق الحوار قد مكن حقل الصفحات،وفيه العدد الأكبر والأقل من الصفحات في المجال "1 – 5"(شاهد الشكل التالي).



إذا ضبط المستخدم هذا الحقل إلى"1-6"سيحدث خطأ يخبر أن المجال هو ضمن "1-5"فقط.ولكن إذا اختار المستخدم جميع الصفحات أو 1-5 أو 1-4 أو 2-3 أو الصفحة الحالية أو أي اختيار،فإن معالج حدث"أطبع صفحة PrintPage " سيتم استدعاءه بنفس الطريقة.في الحقيقة،سيتم استدعاء المعالج كثيراً،وحتى المئات من المرات،حتى تخبره بأن يتوقف.يؤثر اختيار المستخدم في خاصية PrinterSettings.PrintRange وبعض الخاصيات الأخرى،ولكن لايؤثر مباشرةً في عملية الطباعة.إن تغيير سلوك عملية الطباعة المعتمد على هذه الإعدادات راجع إليك.لنتظاهر أن المستخدم أدخل مجال طباعة من 2-3.فلا نستطيع أن نسمح لأداة "طباعة مستند PrintDocument " من إطلاق حدث"أطبع صفحة PrintPage "من أجل كل الصفحات الخمسة،لأنه حتى ولو أنتجنا مخرجات لكل من الصفحة2 و3،سنبقى نحصل على ثلاث صفحات أخرى فارغة تخرج من الطابعة.مانريده هو أن يكون لدينا حدث يتم إطلاقه مرتان فقط.مرة من أجل الصفحة2 ومرةً أخرى من أجل الصفحة3.فنحن بحاجة إلى تعديل أو ضبط استخدام المتغير WhichPageالخاص بالتقاط محتوى الفئة ليكافئ المجال الموافق.أولاً،لنغير معالج حدث"بدء الطباعةBeginPrint "ليستخدم رقم صفحة البدء الصحيح.

```
Private Sub CountingDoc_BeginPrint(ByVal sender As Object, _
ByVal e As System.Drawing.Printing.PrintEventArgs) _
Handles PrintDocument1.BeginPrint
        ' ----- Start the counting over.
        WhichPage = PrintDocument1.PrinterSettings.FromPage
    End Sub
```

في معالج حدث"أطبع صفحة PrintPage "،يجب علينا تعديل الكود الذي يحدد متى تتم مغادرة عملية الطباعة.

```
WhichPage += 1
        If (WhichPage <= PrintDocument1.PrinterSettings.ToPage) Then e.HasMorePages = _ True Else
e.HasMorePages = False
```

بما أن كود المعاينة يتشارك إعدادات نفس المستند،نحتاج لتعديل كود المعاينة لنجبره على طباعة جميع الصفحات دائماً:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
```

```
    ' ----- Display a preview of the document.
    PrintPreviewDialog1.Document = PrintDocument1
    PrintDocument1.PrinterSettings.PrintRange = Printing.PrintRange.AllPages
    PrintPreviewDialog1.ShowDialog()
End Sub
```

إذا شغلت البرنامج وضبطت مجال الطباعة،يجب أن تحصل على الصفحات التي تطلبها فقط.

الطباعة في النمط"الصرف" Printing in "Raw" Mode

إن استخدام GDI+لتوليد صفحات طباعة بسيط جداً.أما من أجل الصفحات المعقدة،من المحتمل أن يكون عليك عمل الكثير من التموضع والقياسات لسلاسل النصوص والترصيف أو الترتيب،ولكن خلاصة القول"ارسم هذا النص أو هذا الشكل عند هذا الموضع".للأسف،لا تدعم كل الطابعات التطبيق الذي يطبع بواسطة GDI وطريقة الطبعة Printish.وهذا بشكل خاص صحيح للطابعات المستخدمة لطباعة كروت التسليف الحرارية thermal credit card . على الرغم من أن بعض هذه الطابعات يمكن أن يكون لها مشغلات على الويندوز،فهي في الحقيقة مصممة لتكون على اتصال مباشر مع تطبيق ما بواسطة لغة " الترشيح أو الفلتر المتسلسل escape sequence " من أجل مثل هذه الطابعات،تحتاج إلى الكتابة بشكل مباشر إلى الطابعة في نمط"المباشر أو الصرف raw " (ما تكتبه من تطبيق خاص تطبعه دون أي معالجة للنص)،حيث أنك تتحكم تماماً بالحروف التي يتم إرسالها إلى الطابعة.(عملياً،ليس عليك الدخول مباشرةً إلى الطابعة،فما يزال بإمكانك الكتابة لطابور(صف)الطابعة printer's queue،وتدع ويندوز من إدارة جدولة عمل الطباعة)

إنه لمن المؤسف أن أخبرك أن الدوت نت ينقصه دعم مثل هذه الطابعات(الطابعات الصرفة أو المباشرة)على الرغم من أن مكتبة الربط الديناميكي DLLيتم تضمينها بنوافذ تمكن طريقة الطباعة المباشرة هذه،فإن دثار دوت نت المدار managed .NET wrapper لهذه المكتبة لايتم شحنه مع إطار العملframework .

حسناً،إن الأمر ليس بهذا السوء،فقد عملت ميكروسوفت وبعض المبرمجين على نشر كود يربط استدعاء مكتبة الربط الديناميكية الغير مدارة unmanaged DLL calls إلى المكافئ المدارmanaged equivalents. سنستخدم تنوع لبعض من هذا الكود في مشروع المكتبة the Library Project في هذا الفصل لدعم طباعة قطع بطاقات صندوق الدفعcheckout slips،أوراق الوصول(فاتورة أو وصلpaper receipts)والتي تمنح الزبون أو المشتري patronمعرفة بالبنود المقيدة للدفع checked out ومتى سيتوجب الدفعdue back .

مشروع Project

كمقدمة،يركز مشروع هذا الفصل على طباعة الشيكات checkoutوفواتير الدفعfinepayment receipts ،ولكن سنضيف أيضاً كل الكود الذي يسمح للمشترين patronsوأمناء المكتبات librarians من تسجيل مدخلات ومخرجات الكتب check in and check out books وبنود المكتبة الأخرى.

دعم الطباعة المباشرة Supporting Raw Printing

إن الكود الأصلي لهذا التطبيق قد أتى من علوم ميكروسوفت المقال الأصلي رقم 322090،والذي يشرح دعم الطباعة المباشرة من تطبيقات الدوت نت.فهو يستخدم ميزة الدوت نت المعروفة بـ" interop " والتي تتيح لكود دوت نت من"التفاعلinteroperate " مع المكونات غير المدارة المعتمدة على الكم COM والتطبيقات الأقدم older unmanaged COM-based components and applications.

أنشئ فئة جديدة وسمها"RawPrinterHelper.vb " واستخدم الكود التالي لتعريفها:

```
Imports System.Runtime.InteropServices
Public Class RawPrinterHelper
    ' الكود في هذه الفئة معتمد على معلومات ميكروسوفت الموضوع القاعدي رقم322090
    '    Web: http://support.microsoft.com/?id=322090
    'API وتصريحات تركيب
    <StructLayout(LayoutKind.Sequential, CharSet:=CharSet.Unicode)> Private Structure DOCINFOW
        <MarshalAs(UnmanagedType.LPWStr)> Public pDocName As String
        <MarshalAs(UnmanagedType.LPWStr)> Public pOutputFile As String
        <MarshalAs(UnmanagedType.LPWStr)> Public pDataType As String
    End Structure
    <DllImport("winspool.Drv", EntryPoint:="OpenPrinterW", SetLastError:=True,
CharSet:=CharSet.Unicode, _
       ExactSpelling:=True, CallingConvention:=CallingConvention.StdCall)> _
    Private Shared Function OpenPrinter(ByVal src As String, ByRef hPrinter As IntPtr, ByVal pd As
Long) As Boolean
    End Function
    <DllImport("winspool.Drv", EntryPoint:="ClosePrinter", SetLastError:=True,
CharSet:=CharSet.Unicode, _
       ExactSpelling:=True, CallingConvention:=CallingConvention.StdCall)> _
    Private Shared Function ClosePrinter(ByVal hPrinter As IntPtr) As Boolean
    End Function
    <DllImport("winspool.Drv", EntryPoint:="StartDocPrinterW", SetLastError:=True,
CharSet:=CharSet.Unicode, _
       ExactSpelling:=True, CallingConvention:=CallingConvention.StdCall)> _
    Private Shared Function StartDocPrinter(ByVal hPrinter As IntPtr, ByVal level As Int32, ByRef
pDI As DOCINFOW) As Boolean
    End Function
    <DllImport("winspool.Drv", EntryPoint:="EndDocPrinter", SetLastError:=True,
CharSet:=CharSet.Unicode, _
```

```
        ExactSpelling:=True, CallingConvention:=CallingConvention.StdCall)> _
    Private Shared Function EndDocPrinter(ByVal hPrinter As IntPtr) As Boolean
    End Function
    <DllImport("winspool.Drv", EntryPoint:="StartPagePrinter", SetLastError:=True,
CharSet:=CharSet.Unicode, _
        ExactSpelling:=True, CallingConvention:=CallingConvention.StdCall)> _
    Private Shared Function StartPagePrinter(ByVal hPrinter As IntPtr) As Boolean
    End Function
    <DllImport("winspool.Drv", EntryPoint:="EndPagePrinter", SetLastError:=True,
CharSet:=CharSet.Unicode, _
        ExactSpelling:=True, CallingConvention:=CallingConvention.StdCall)> _
    Private Shared Function EndPagePrinter(ByVal hPrinter As IntPtr) As Boolean
    End Function
    <DllImport("winspool.Drv", EntryPoint:="WritePrinter", SetLastError:=True,
CharSet:=CharSet.Unicode, _
        ExactSpelling:=True, CallingConvention:=CallingConvention.StdCall)> _
    Private Shared Function WritePrinter(ByVal hPrinter As IntPtr, ByVal pBytes As IntPtr, ByVal
dwCount As Int32, ByRef dwWritten As Int32) As Boolean
    End Function
    Public Shared Function SendStringToPrinter(ByVal targetPrinter As String, ByVal stringContent
As String, ByVal documentTitle As String) As Boolean
        ' إرسال مصفوفة بايت إلى طابور الطابعة.العودة بصواب في حال النجاح
        Dim printerHandle As IntPtr
        Dim errorCode As Int32
        Dim docDetail As DOCINFOW = Nothing
        Dim bytesWritten As Int32
        Dim printSuccess As Boolean
        Dim contentBytes As IntPtr
        Dim contentSize As Int32
        On Error Resume Next
        ' تثبيت الفراغات في بداية أسطر هذا المستند
        With docDetail
            .pDocName = documentTitle
            .pDataType = "RAW"
        End With
        ' تحويل النص إلى نص آنسي
        ' ANSI text
        contentSize = stringContent.Length()
        contentBytes = Marshal.StringToCoTaskMemAnsi(stringContent)
        ' فتح الطابعة وطباعة المستند
        printSuccess = False
        If OpenPrinter(targetPrinter, printerHandle, 0) Then
            If StartDocPrinter(printerHandle, 1, docDetail) Then
                If StartPagePrinter(printerHandle) Then
                    ' إرسال المحتويات إلى الطابعة
                    printSuccess = WritePrinter(printerHandle, contentBytes, _
                        contentSize, bytesWritten)
                    EndPagePrinter(printerHandle)
                End If
                EndDocPrinter(printerHandle)
            End If
            ClosePrinter(printerHandle)
        End If
        'GetLastError يوفر
        ' معلومات حول الخطأ الأخير,حاليا تجاهله
        If (printSuccess = False) Then errorCode = Marshal.GetLastWin32Error()
        ' تحرير الذاكرة الغير مستخدمة
        Marshal.FreeCoTaskMem(contentBytes)
        ' الإتمام
        Return printSuccess
    End Function
End Class
```

إن الكود نسبياً واضح المعالم clear-cut ،حيث أن الطريقة SendStringToPrinterتحضر نص ما للطباعة بتحويله عنوة إلى هيئة أنسي القياسي standard ANSI.ومن ثم يستخدم الوظائف(الدوال) في مكتبة *winspool.drv* لفتح عمل طباعة جديد،ويرسل المحتوى المحضر إليها.على العموم يوجد الكثير من الترتيب(الترتيب المتصاعدmarshalling) المستمر في الكود من خلال أعضاء فئة Marshal class.حيث أن مكتبة

winspool.drv غير مدارة.جميع البيانات يجب أن تنقل جيئة وذهاباً بشكل غير مباشر بين تطبيق المكتبة المدارة ومكتبة *winspool.drv* الغير مدارة .

طباعة البطاقات(التذاكر)Printing Tickets

والآن لدينا فئة مريحة بحيث يمكننا إرسال أي محتوى مباشر إلى أي طابعة خاصة،لنضيف بعض الكود لكي نستخدمها.أولاً،نحتاج إلى إضافة فئة مساعدة من أجل طباعة تذكرة خاصة بزبون ما.أنشئ فئة جديدة وسمها CheckedOutItem.vb وبدل محتواها الفارغ بالكود **التالي**:

```
Public Class CheckedOutItem
   ' تستخدم لتخزين كل بند تم إخراجه على الفورم الرئيسية
   ' على الرغم من أنها تدعم طباعة وصول الاستلام
    Public ItemTitle As String
    Public CopyNumber As Integer
    Public Barcode As String
    Public DueDate As Date
End Class
```

سنستخدم هذه الفئة لنقل التفاصيل لأن يتم طباعتها على الفاتورة(التذكرة) عندما يتم تفحص بنود المخرجات.فيما يخص طباعة التذاكر،لنعمل على إضافة الفئة التي تقوم بعملية الطباعة الحقيقية. قم بإنشاء ملف وحدة برمجية جديد module (وليس فئة) وسمه *TicketPrinting.vb* وأضف له كود الذي يملأ الوحدة البرمجية :

```
Module TicketPrinting
...
End Module
```

يحتوي الكود على ثلاث طرق تقود عملية الطباعة: PrintCheckoutTicket و PrintBalanceTicket و PrintPaymentTicket.يتم استدعاء هذه الطرق من أجزاء أخرى ضمن التطبيق عندما يحين وقت تقديم كرت مطبوع للمستخدم.تتضمن الوحدة البرمجية أيضاً العديد من الطرق الأخرى والتي تدعم هذه الطرق الرئيسية الثلاث.وحيث أن هذه الطرق الثلاث نوعاً ما متشابهة في التركيب،لذلك لنلقي نظرة على الدالة PrintCheckoutTicket.

```
Public Sub PrintCheckoutTicket(ByVal patronID As Integer, _
            ByVal checkedOutItems As ListBox)
        ' ----- Print out a ticket of what the patron checked out. The supplied
        '       ListBox control contains objects of type CheckedOutItem.
        Dim ticketWidth As Integer
        Dim ticketText As System.Text.StringBuilder
        Dim counter As Integer
        Dim patronFines As Decimal
        Dim itemDetail As CheckedOutItem

        On Error GoTo ErrorHandler

        ' ----- Ignore if there is nothing to print.
        If (patronID = -1) Or (checkedOutItems.Items.Count = 0) Then Return

        ' ----- Get the width of the ticket.
        ticketWidth = My.Settings.ReceiptWidth
        If (ticketWidth <= 0) Then ticketWidth = 40

        ' ----- Build the heading.
        ticketText = GetTicketHeader(patronID, ticketWidth)
        If (ticketText Is Nothing) Then Return

        ' ----- Process each checked-out item.
        For counter = 0 To checkedOutItems.Items.Count - 1
            ' ----- Extract the detail from the list.
            itemDetail = CType(checkedOutItems.Items(counter), CheckedOutItem)

            ' ----- Add the item name.
            ticketText.AppendLine(Left(itemDetail.ItemTitle, ticketWidth))

            ' ----- Add the bar code number and due date.
            ticketText.AppendLine(LeftAndRightText(itemDetail.Barcode, _
                "Due: " & Format(itemDetail.DueDate, "MMM d, yyyy"), ticketWidth))
            ticketText.AppendLine()
        Next counter

        ' ----- If there are fines due, print them here.
        patronFines = CalculatePatronFines(patronID)
        If (patronFines > 0@) Then
            ticketText.AppendLine("Fines Due: " & Format(patronFines, "Currency"))
            ticketText.AppendLine()
        End If

        ' ----- Add the bottom display text.
        ticketText.Append(GetTicketFooter(ticketWidth))
```

```
        ' ----- Send the ticket to the printer.
        RawPrinterHelper.SendStringToPrinter(My.Settings.ReceiptPrinter, _
            ticketText.ToString(), "Checkout Receipt")
        Return

ErrorHandler:
        GeneralError("TicketPrinting.PrintCheckoutTicket", Err.GetException())
        Return
    End Sub
```

يبني الكود متغير نصي ما(فعلياً هو باني النصStringBuilder) لعرض المحتوى،إضافة تفاصيل حول كل بند اختبار مخرج لمجزئ النصوص string buffer.ومن ثم يقوم باستدعاء SendStringToPrinterلإرسال المحتوى لطابعة التذاكر(الفواتير receipt printer) المركبة(My.Settings.ReceiptPrinter).سنقوم بإضافة الكود الذي يستدعي PrintCheckoutTicketفيما بعد،أما الآن لنضيف الكود الذي يستدعي الطريقتين الأخريين .عندما يغلق نموذج سجل المدفوعاتPayment Record form ،نريد أن نطبع تذكرة لجميع المدفوعات المعمولة بشكل آلي في الوقت الذي يتم فيه فتح النموذج.قم بإضافة الكود التالي إلى معالج حدثPatronRecord.ActClose_Click ،تماماً قبل الكود الموجود حالياً في المعالج.

```
Private Sub ActClose_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ActClose.Click
' طباعة بطاقة عند الحاجة
        If (SessionPayments.Count > 0) Then _
            PrintPaymentTicket(ActivePatronID, SessionPayments)
        SessionPayments.Clear()
        SessionPayments = Nothing
' إغلاق الفورم
        Me.DialogResult = Windows.Forms.DialogResult.Cancel
    End Sub
```

طباعة أكواد التعريف(التي تعرف الكمبيوتر على أدوات معينة):

يطبع مشروع المكتبة ثلاث أنواع من أكواد التعريف:(1)بند أكواد التعريف التي يمكنك أن تلصقها على الكتب،السيديات،وأي شيء أخر يمكن أن يتم أخراجه أو يتم إدارته بواسطة النظام،(2):أكواد تعريف المشتري والذي يمكن عمله ضمن كروت المشتريين المتطابقة،(3):أكواد تعريف متنوعة(الأخرى) والتي يمكن للمكتبة ن تستخدمه لأي هدف أخر.كل من هذه أكواد التعريف الثلاث يتم طباعتها من خلال النموذج BarcodePrint.يبين الشكل التالي الأدوات المحتواة ضمن هذه الفورم.
لقد عملت مسبقاً على إضافة هذه الفورم إلى المشروع،مع كودها،إليك الكود الخاص بزر المعاينة،والذي سيبدو معروفاً بعد أن أقدمه لك من خلال هذا الفصل.

```
Private Sub ActPrint_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ActPrint.Click
' يريد المستخدم طباعة اللصاقات
        Dim whichTarget As Integer
        On Error Resume Next
' التأكد من أن المستخدم قد زود بيانات صحيحة
        If (VerifyFields() = False) Then Return
' تحميل جميع التفاصيل الخاصة بصفحة لأن يتم استخدامها في الطباعة
        If (LoadPageDetails() = False) Then Return
' السؤال قبل طباعة اللصاقات ومن ثم الطباعة.
        Me.Cursor = Windows.Forms.Cursors.WaitCursor
        PageSoFar = 0
        PreviewMode = False
        BarcodeDoc = New System.Drawing.Printing.PrintDocument
        BarcodePrintDialog.Document = BarcodeDoc
        If (BarcodePrintDialog.ShowDialog() <> Windows.Forms.DialogResult.OK) Then
            BarcodeDoc = Nothing
            Me.Cursor = Windows.Forms.Cursors.Default
            Return
        End If
        BarcodeDoc.Print()
        BarcodeDoc = Nothing
        Me.Cursor = Windows.Forms.Cursors.Default
' سؤال المستخدم حول نجاح الطباعة
        If (MsgBox("هل تمت طباعة اللصاقات بنجاح؟", MsgBoxStyle.YesNo Or MsgBoxStyle.Question,
ProgramTitle) = _
            MsgBoxResult.No) Then Return
' تحديث قاعدة البيانات بقيم الطباعة
        whichTarget = CInt(CType(RecordType.SelectedItem, ListItemData))
        If (whichTarget = BarcodeOutput.Item) Then
' لصاقات بند
            SetSystemValue("NextBarcodeItem", CStr(Val(RecordStop.Text) + 1))
```

```
        ElseIf (whichTarget = BarcodeOutput.Patron) Then
' لصاقات زبون
            SetSystemValue("NextBarcodePatron", CStr(Val(RecordStop.Text) + 1))
        Else
' لصاقات متنوعة
            SetSystemValue("NextBarcodeMisc", CStr(Val(RecordStop.Text) + 1))
        End If
' إغلاق الفورم
        ActClose.PerformClick()
    End Sub
```



كود زر الطباعة مشابه لهذا الكود تقريباً،ولكنه يستخدم حالة PrintDialogبدل PrintPreviewDialog.ويتعقب أيضاً رقم كود التعريف النهائي، بالتالي يمكن أن يساعد على تجنب تداخل طباعتها في المرة القادمة.
يعمل معالج الحدث BarcodeDoc_PrintPage طباعة كود التعريف الحقيقي.يعمل كوده على دمج معالجي حدث BarcodeLabel.PreviewArea_Paint و BarcodePage.PreviewArea_Paint إلى جهاز طباعة رائع واحد.لتمكين استخدام نموذج طباعة كود التعريف،أضف العبارات التالية إلى معالج الحدث ActReportsBarcode_Clickفي فئة الفورم الرئيسية MainForm.

```
Private Sub ActReportsBarcode_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
ActReportsBarcode.Click
        ' التأكد من أن عمل التالي مسموح للمستخدم
        If (SecurityProfile(LibrarySecurity.ManageBarcodeTemplates) = False) Then
            MsgBox(NotAuthorizedMessage, MsgBoxStyle.OkOnly Or MsgBoxStyle.Exclamation,
ProgramTitle)
            Return
        End If
        ' إظهار نموذج طباعة لصاقة كود التعريف
        Call (New BarcodePrint).ShowDialog()
    End Sub
```

تجديد بنود الزبون المخرجة.Renewal of Checked-Out Patron Items

من أجل زبون المكتبة،الشيء الوحيد الأكثر أهمية من البنود المخرجة والمدخلة هو القدرة على قراءة تلك البنود.لن يساعد مشروع المكتبة أياً كان بذلك،ولكنه سيعمل أشياء مداولة الإخراج،الإدخال على طول الكود الذي نضيفه في هذا الفصل.لنعمل على إضافة كود التجديد من أجل البنود المخرجة حالياً.الزر"تجديد Renew " على نموذج سجل الزبون Patron Recordيبدأ هذه العملية.أضف الكود إلى معالج حدث النقر PatronRecord.ActRenewItemsOut_Clickوالذي يعمل التجديد.

```
Private Sub ActRenewItemsOut_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ActRenewItemsOut.Click
' محاولة تجديد بند تم إخراجه
        Dim sqlText As String
        Dim dbInfo As SqlClient.SqlDataReader
        Dim checkOutDate As Date
        Dim dueDate As Date
        Dim renewsSoFar As Integer
        Dim maxRenew As Integer
        Dim renewDays As Integer
        Dim itemDetail As PatronDetailItem

        On Error GoTo ErrorHandler
' التأكد من أن البند تم اختياره
        If (ItemsOut.SelectedIndex = -1) Then Return
        itemDetail = CType(ItemsOut.SelectedItem, PatronDetailItem)
' التأكد من أن هذا البند غير مرجعي
        sqlText = "SELECT IC.Available, IC.Reference FROM PatronCopy AS PC " & _
            "INNER JOIN ItemCopy AS IC ON PC.ItemCopy = IC.ID " & _
            "WHERE PC.ID = " & itemDetail.DetailID
        dbInfo = CreateReader(sqlText)
        dbInfo.Read()
        If (CBool(dbInfo!Available) = False) Then
```

```
            dbInfo.Close()
            dbInfo = Nothing
            MsgBox("You cannot renew this item because it is no longer available " & _
                "for circulation or checkout.", MsgBoxStyle.OkOnly Or _
                MsgBoxStyle.Exclamation, ProgramTitle)
            Return
        End If
        If (CBool(dbInfo!Reference) = True) Then
            dbInfo.Close()
            dbInfo = Nothing
            MsgBox("You cannot renew this item because it is a reference item.", _
                MsgBoxStyle.OkOnly Or MsgBoxStyle.Exclamation, ProgramTitle)
            Return
        End If
        dbInfo.Close()
' الحصول على حالة بند تم إخراجه
        sqlText = "SELECT CheckOut, DueDate, Renewal FROM PatronCopy WHERE ID = " & _
            itemDetail.DetailID
        dbInfo = CreateReader(sqlText)
        dbInfo.Read()
        checkOutDate = CDate(dbInfo!CheckOut)
        dueDate = CDate(dbInfo!DueDate)
        renewsSoFar = CInt(dbInfo!Renewal)
        dbInfo.Close()
' تحديد الحدود على هذا النوع من البند
        sqlText = "SELECT * FROM CodeMediaType WHERE ID = (SELECT NI.MediaType " & _
            "FROM PatronCopy AS PC INNER JOIN ItemCopy AS IC ON PC.ItemCopy = IC.ID " & _
            "INNER JOIN NamedItem AS NI ON IC.ItemID = NI.ID " & _
            "WHERE PC.ID = " & itemDetail.DetailID & ")"
        dbInfo = CreateReader(sqlText)
        dbInfo.Read()
        maxRenew = CInt(dbInfo!RenewTimes)
        renewDays = CInt(dbInfo!RenewDays)
        dbInfo.Close()
        dbInfo = Nothing
' رؤية فيما إذا قد تم تجديده أيضاً
        If (renewsSoFar >= maxRenew) Then
            If (maxRenew = 0) Then
                MsgBox("لاتستطيع تجديد هذا النوع من البند لأن هذا النوع من البند " & "غير مرخص تجديده.",
MsgBoxStyle.OkOnly Or _
                    MsgBoxStyle.Exclamation, ProgramTitle)
            Else
                MsgBox("لاتستطيع تجديد هذا البند لأنك جددته للتو " & CStr(IIf(renewsSoFar = 1, "1
time", renewsSoFar & " times")) & ",البنود من النوع لهذا والحدود", MsgBoxStyle.OkOnly Or _
                    MsgBoxStyle.Exclamation, ProgramTitle)
            End If
            Return
        End If
        Me.Cursor = Windows.Forms.Cursors.WaitCursor
' تجديد هذا البند بضبط تاريخ استحقاق الإعادة.بينما هناك مساحة
' حافظ على تجديد البند حتى لايتجاوز البند استحقاق الإعادة,عند الحاجة
        Do While True
            dueDate = DateAdd(DateInterval.Day, renewDays, dueDate)
            renewsSoFar += 1
            If (renewsSoFar >= maxRenew) Then Exit Do
            If (dueDate >= Today) Then Exit Do
        Loop
' لاتسمح لتاريخ الاستحقاق أن يكون يوم عطلة
        Do While (IsHolidayDate(dueDate) = True)
            dueDate = DateAdd("d", 1, dueDate)
        Loop
        TransactionBegin()
' تحديث السجل
        sqlText = "UPDATE PatronCopy SET DueDate = " & DBDate(dueDate) & _
            ", Renewal = " & renewsSoFar & " WHERE ID = " & itemDetail.DetailID
        ExecuteSQL(sqlText)
' تحديث سجل الزبون
        sqlText = "UPDATE Patron SET LastActivity = GETDATE() WHERE ID = " & ActivePatronID
        ExecuteSQL(sqlText)
```

```
        TransactionCommit()
'تحـديث العـرض
        itemDetail.DueDate = dueDate
        ItemsOut.Invalidate(ItemsOut.GetItemRectangle(ItemsOut.SelectedIndex))
        Me.Cursor = Windows.Forms.Cursors.Default
        Return
ErrorHandler:
        Me.Cursor = Windows.Forms.Cursors.Default
        GeneralError("PatronRecord.ActRenewItemsOut_Click", Err.GetException())
        On Error Resume Next
        If Not (dbInfo Is Nothing) Then dbInfo.Close() : dbInfo = Nothing
        TransactionRollback()
        Return
    End Sub
```

يعمل الكود بعض الحسابات لتحديد تاريخ استحقاق الإعادة الجديد(متجنباً العطل) ومن ثم تحديث قاعدة البيانات في المداولة.

```
TransactionBegin()
...
```

دعم الإعادة والإخراج.Support for Check-In and Checkout

إذا أضافت مكتبة لصاقات كود التعريف لجميع بنودها،فسيكون الإدخال والإخراج بواسطة قارئ كود التعريف.ولكن بالنسبة لمكتبة صغيرة تستخدم البرنامج يمكن أن لا يحتاج الكادر دعم الإخراج والإدخال بالعنوان.خلال الإخراج أو الإدخال،إما يدخل المستخدم كود التعريف أو العنوان(جزئي أو كامل). و من المفترض أن لا تكون المدخلات العددية عناوين،ولبدء البحث بالعنوان.الفورم CheckLookup.vb الجديد،المبين في الشكل التالي يعرض جميع التطابقات من أجل عنوان تم إدخاله.



على الرغم من أن الحقول على الفورم بشكل أولي مطابقة لما تم إخراجه فقط،تقوم الفورم بمهام مضاعفة،تحويل مظهرها من أجل أهداف الإدخال.بالإضافة إلى جدولة المدخلات محدودة إلى تلك البنود المتطابقة والتي تم إخراجها سابقاً.
عملت على إضافة هذه الفورم لمشروعك،مع الكود المصدري.يستعلم معظم الكود عن قاعدة البيانات من أجل البنود المتطابقة ويعرض النتائج باستخدام حامل رسم صندوق القائمةowner draw listbox. وهو مجموعة جزئية من الكود الموجود في الفورمItemLookup.vb. الاختلاف الحقيقي الوحيد بين المدخلات والمخرجات يحدث في الطريقة PerformLookup.يبدأ مقطع الكود هذا ببناء أمر سكول الاختيار بند رئيسي،ومن ثم إنهائه بهذه العبارة.

```
If (asCheckIn) Then sqlText &= " AND IC.ID IN" _
              Else sqlText &= " AND IC.ID NOT IN"
        sqlText &= " (SELECT ItemCopy FROM PatronCopy WHERE Returned = 0)"
```

لذلك فالاختلاف في " IN "مقابل " NOT IN ".
الدالة CheckItemByTitleهي الواجهة الرئيسية لمنطق الفورم.

```
Public Function CheckItemByTitle(ByVal CheckIn As Boolean, ByVal searchText As String) As Integer
```

تمرر لهذه الدالة العنوان المزود من قبل المستخدم(searchText)وعلامة تشير إلى حالة الإدخال أو الإخراج،وتعود بحقل قاعدة البيانات the ItemCopy.ID من أجل بند المكتبة المختار.
جميع التغيرات في هذا الفصل تحدث في فئة الفورم الرئيسيةMainForm،لذلك دعنا نذهب إليها.تضبط الطريقة UpdateDisplayForUserميزات الفورم الرئيسية عندما يدخل المدير أو يخرج.واحدة من الميزات التي لم نأخذها بالحسبان من قبل، وهي إمكانية المدراء تعريف إخراجات بنود من تحديدهم دون مساعدة أمناء المكتبة.لدعم هذه الميزة،نحتاج إلى تغيير بعض الكود في الطريقةUpdateDisplayForUser . بدل الأسطر التالية من الكود:

```
LabelTasks.Visible = False
LineTasks.Visible = False
PicCheckOut.Visible = False
ActCheckOut.Visible = False
```

بالأسطر التالية :

```
'لنرى فيما إذا بإمكان الزبائن إخراج بنود بأنفسهم
        Dim userCanCheckOut As Boolean = _
        CBool(Val(GetSystemValue("PatronCheckOut")))
        LabelTasks.Visible = userCanCheckOut
        LineTasks.Visible = userCanCheckOut
        PicCheckOut.Visible = userCanCheckOut
        ActCheckOut.Visible = userCanCheckOut
```

ونحتاج أيضاً إلى إضافة كود أمن مناسب مشابه للطريقة TaskCheckOut.إليك الأسطر الأولى من الكود من تلك الطريقة .

```
'تحديث العرض
        AllPanelsInvisible()
        If (SecurityProfile(LibrarySecurity.CheckOutItems)) Then _
            PanelCheckOut.Visible = True
```

بدله بالكود التالي:

```
'نمط الإخراج أو الإعارة
        Dim userCanCheckOut As Boolean
'لنرى فيما إذا بإمكان الزبائن إخراج بنود بأنفسهم
        userCanCheckOut = CBool(Val(GetSystemValue("PatronCheckOut")))
'  تحديث العرض
        AllPanelsInvisible()
        If (userCanCheckOut Or _
            SecurityProfile(LibrarySecurity.CheckOutItems)) Then _
            PanelCheckOut.Visible = True
```

إخراج البنود الحقيقي يحدث على الفورم الرئيسية MainFormنفسها.أولاً،يحدد الزبون،ومن ثم يتم معالجة البند الذي سيتم إخراجه.دعنا نضيف متغير على مستوى الفئة إلى كود الفورم الرئيسية لتتبع الزبون.وطالما أننا أضفنا التعريفات،سنعمل أيضاً على إضافة ثوابت تشير إلى الصور المخزنة في الأداةMainForm.StatusImages. سيتم استخدام هذه الثوابت في بعض الكود ذو الصلة بالإدخال المضاف فيما بعد.أضف الكود التالي إلى تعريف الفئة.

```
Private ActiveCheckOutPatron As Integer = -1
    Private Const StatusImageBad As Integer = 0
    Private Const StatusImageGood As Integer = 1
```

عندما يحدد المستخدم الزبون للاستخدام من أجل الإخراج،ومن ثم يبدأ الإدخال إخراج البنود،ستكون الخطوة الرئيسية هي النقر على الزر إنهاء،مشيراً إلى نهاية عملية الإخراج من أجل زبون.(شاهد الشكل التالي إذا كنت تريد رؤية زر "إنهاء"الآن).مهما يكن،لايوجد شيء يعمل على إيقاف المستخدم من القفز إلى جزء آخر من البرنامج،أو من الخروج من البرنامج بالكامل،بدون نقر الزر "إنهاء".بدون النقر أولاً على الزر "إنهاء " أولاً،يجب أن نستبق هذا السلوك الغير مرغوب بحيث يتوافق مع برمجيات المستخدمين.لضمان إتمام الإخراج بشكل مناسب،سنعمل على إضافة بعض الكود إلى ثلاث أماكن في الفورم الرئيسية والذي سيعمل على إقتناص أي فعل غير مرغوب من قبل المستخدم. أضف الكود التالي إلى بداية الطرق الثلاث التالية:(1) معالج الحدث MainForm_FormClosing،(2)الطريقة ShowLoginForm،و(3) الطريقة AllPanelsInvisible.

```
'إنهاء عمليات الإخراج عند الحاجة
        If (ActiveCheckOutPatron <> -1) Then ActFinishCheckOut.PerformClick()
```

البنود المخرجة.Checking Out Items

يظهر جميع كود الإخراج في فئة الفورم الرئيسية(ما عدا الكود في الفورم *CheckLookup.vb*).الإخراجات واحدة من ثمانية لوحات عرض رئيسية يمكن الوصول لها من خلال هذه الفورم.(شاهد الشكل التالي).إليك المعالجة من أجل البنود المخرجة من لوحة "البنود المعارة":

1. ينقر المستخدم زر"زبون" ويحدد الزبون الذي سيخرج البنود.
2. يعمل المستخدم على إدخال العنوان أو كود التعريف من أجل كل بند من أجل إخراجه،وينقر الزر "إخراج" من أجل كل واحد.
3. ينقر المستخدم الزر"إنهاء" عند يكتمل الإخراج.



لنعمل على إضافة الكود من أجل كل من هذه الأزرار الثلاث.أولاً،أضف الكود من أجل معالج ActCheckOutPatron_Click.
يطلب هذا لكود من المستخدم من أجل اختيار الزبون،ويعرض الحقول المتبقية في حال النجاح.إليك جزء الكود الذي يعرض سؤال المستخدم:

```
        ' ----- Get the ID of the patron.
        patronID = (New PatronAccess).SelectPatron()
        If (patronID = -1) Then Return
        ' ----- Get the patron name.
        sqlText = "SELECT FirstName + ' ' + LastName FROM Patron " & _
            "WHERE ID = " & patronID
        patronName = CStr(ExecuteSQLReturn(sqlText))
        ' ----- Is this patron active?
        sqlText = "SELECT Active FROM Patron WHERE ID = " & patronID
        If (CBool(ExecuteSQLReturn(sqlText)) = False) Then
            MsgBox("Patron '" & patronName & "' is marked as inactive.", _
                MsgBoxStyle.OkOnly Or MsgBoxStyle.Exclamation, ProgramTitle)
            Return
        End If
```

أضف كود إلى معالج الحدثActDoCheckOut_Click، والذي يعالج كل بند من خلال الزر"إخراج".
كما ذكرت من قبل،يفرق هذا الكود بين المدخلة العددية(أكواد التعريف) والمدخلات الأخرى(العناوين).

```
If (IsNumeric(Trim(CheckOutBarcode.Text))) Then
            ' ----- Probably a barcode supplied. Get the related ID.
            sqlText = "SELECT ID FROM ItemCopy WHERE Barcode = " & _
                DBText(Trim(CheckOutBarcode.Text))
            copyID = DBGetInteger(ExecuteSQLReturn(sqlText))
            If (copyID = 0) Then
                ' ----- Invalid barcode.
                MsgBox("Barcode not found.", _
                    MsgBoxStyle.OkOnly Or MsgBoxStyle.Exclamation, ProgramTitle)
                CheckOutBarcode.Focus()
                CheckOutBarcode.SelectAll()
                Return
            End If
```

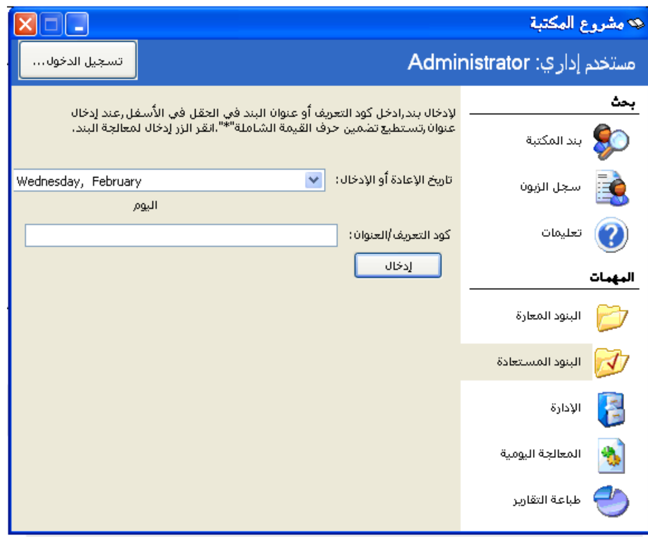
أخيراً،بعد التحقق من أن البند متاح للاستخدام،يعمل الكود على إخراج البند بتحديث سجلات قاعدة البيانات ذات العلاقة.

```
TransactionBegin()
        ' ----- Update patron copy record.
        sqlText = "INSERT INTO PatronCopy (Patron, ItemCopy, CheckOut, Renewal, " & _
            "DueDate, Returned, Missing, Fine, Paid) VALUES (" & _
            ActiveCheckOutPatron & ", " & copyID & ", " & DBDate(Today) & _
            ", 0, " & DBDate(untilDate) & ", 0, 0, 0, 0)"
        ExecuteSQL(sqlText)
```

الزر الثالث والأخير هو الزر"إنهاء".أضف الكود إلى معالج الحدث ActFinishCheckOut_Click.
يعمل هذا الكود على إعادة تنضيد حقول العرض في التحضير إلى الإخراج التالي للزبون.
صندوق القائمة listboxعلى لوحة"البنود المعارة" يحتاج إلى عرض عمودين من البيانات:(1)تاريخ الإعادة،و(2)تفاصيل البند،مثل العنوان وكود التعريف.هذه القيم تم إضافتها إلى القائمة باستخدام الفئة CheckedOutItemالتي أضفناها سابقاً في هذا الفصل.أضف الكود إلى معالج حدث CheckedOutItems_DrawItem.

البنود المدخلة.Checking In Items

البنود المدخلة أسهل بكثير بما أننا لا نحتاج إلى تحديد الزبون أولاً. كود التعريف أو عنوان البند المدخل كافي لإتمام جميع العمليات. يبين الشكل التالي لوحة "البنود المستعادة".



تتضمن هذه اللوحة مؤشر التاريخ عندما سيتم إعادة البند.عادةً،هو اليوم الحالي،ولكن إذا تم إعادة بنود المكتبة إلى مخزن المكتبة خلال الليل بعد ساعات العمل،من المحتمل أن أمين المكتبة يريد ضبط التاريخ إلى "البارحة"،فقط في حال تم إعادة هذه البنود قبل منتصف الليل.دعنا نضيف بعض الكود بحيث تشير اللوحة إلى "اليوم" أو "البارحة" أو إلى يوم أخر عند تغيير التاريخ.أضف الكود التالي إلى معالج الحدث CheckedInDate_ValueChanged.

```
Private Sub CheckInDate_ValueChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles
CheckInDate.ValueChanged
        ' ضبط يوم الإدخال على العرض
        Select Case DateDiff(DateInterval.Day, CheckInDate.Value, Today)
            Case 0     ' اليوم
                CheckInDay.Text = "اليوم"
                CheckInDay.BackColor = SystemColors.Control
                CheckInDay.ForeColor = SystemColors.ControlText
            Case 1     ' البارحة
                CheckInDay.Text = "البارحة"
                CheckInDay.BackColor = Color.Red
                CheckInDay.ForeColor = Color.White
            Case Else ' يوم ما مضى
                CheckInDay.Text = DateDiff(DateInterval.Day, _
                    CheckInDate.Value, Today) & " الأيام الماضية"
                CheckInDay.BackColor = Color.Red
                CheckInDay.ForeColor = Color.White
        End Select
    End Sub
```

الإدخال الفعلي يحدث عندما يعمل المستخدم على إدخال الباركود(كود التعريف) في حقل النص،وينقر على زر "إدخال".أضف معالج حدث ActDoCheckIn_Clickللفورم الرئيسية.

```
Private Sub ActDoCheckIn_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
ActDoCheckIn.Click
...
```

بعد عمل بعض البحث وتأكيد المراجعة،يعمل الكود على إدخال البند من خلال تحديث قاعدة البيانات.

```
'عمل إدخال في المداولة
        TransactionBegin()
' تحديث سجل نسخة الزبون
        sqlText = "UPDATE PatronCopy SET CheckIn = " & DBDate(CheckInDate.Value) & _
            ", Returned = 1 WHERE ID = " & patronCopyID
        ExecuteSQL(sqlText)
' تحديث سجل الزبون
        sqlText = "UPDATE Patron SET LastActivity = GETDATE() WHERE ID = " & patronID
        ExecuteSQL(sqlText)
        TransactionCommit()
```

هذا كل شيء الآن،ما تبقى إضافة الكود الذي يعالج بشكل مناسب غرامات البنود قبل إدخالها،وسنعمل ذلك في الفصل 22.

التقارير Reporting

من أجل مطوري تطبيقات العمل،التقارير هي حقيقة الحياة.من المحتمل أنك تمضي وقتك في تطوير واجهات المستخدم أو استكشاف وفهم الخوارزميات المستخدمة بشكل عام في مبادئ احتساب البيانات المقبولة،ولكن بالمقابل،تستثمر العديد من الساعات المملة كل أسبوع في إنتاج تقرير بعد تقرير.وتشكل هذه التقارير رسوم مهمة على طائفة البرمجة. في أمريكا لوحدها، تقدر مراكز التحكم والوقاية من الأمراض حوالي 850تقرير متعلق بالأموات سنوياً-وهذا لا يتضمن حتى إحصاء الذين قرؤوا تلك التقارير. لذلك إذا كنت مبرمج عملي،فإن التقارير داخل مستقبلك.تتضمن الفيجوال أستوديو و الدوت نت العديد من الميزات المتمركزة حول التقرير وأدوات يمكنك استخدامها تماماً بلا حدود.
يناقش هذا الفصل بعضاً من موارد هذه التقارير،ويحاول بجهد العثور على معلومات أعمق حول أدوات التقرير المستخدمة في مشروع المكتبة.

خيارات التقرير في الدوت نت.Report Options in .NET

يتضمن التقرير عرض وطباعة بيانات أساسية ومختصرة للمستخدم من أجل أهداف عملية معينة.تتضمن الفيجوال بيسك2008بطبعتها الاحترافية ست طرق رئيسية لإتمام هذا الهدف.بعض الطبعات الأخرى تضيف إلى أو تقلل من مجموعة الخيارات هذه،وبإمكانك دائماً تحسين هذه القائمة باستخدام أدوات مشاركة ثانوية.

الطباعة المعتمدة على"طباعة مستند PrintDocument " PrintDocument-Based Printing.

كما تعلمنا في الفصل السابق،يتضمن إطار عمل الدوت نت نظام طباعة معتمد على كائن objectbased يستخدم أوامر GDI+لرسم نص ورسوميات graphicsعلى كل صفحة طباعة.بما أنك تستطيع وضع أي شيء تريده على كل صفحة،فتستطيع تطوير تقاريرك الخاصة باستخدام هذه الطريقة.إن مسؤولية وضع كل لافتة وحقول محسوبة على الصفحة،وتحديد متى سيتم التنقل إلى صفحة جديدة،كل هذا بالكامل سيكون على عاتقك.ما تزال أوامر الـGDI+ بسيطة،وتطوير بعض التقارير القاعدية باستخدام هذه الطريقة لن يكون محبباً.إذا كنت تريد أخذ هذا المسلك من أجل تقاريرك،أفضل عودتك إلى الفصل20 ومبادئ الطباعة الأساسية الموجودة هناك.

صفحات HTML / الويب. HTML/Web Pages

إن الانترنت(ولغة وصف الصفحات المعتمدة على HTML (لغة ترميز النصوص الفائقة))وسيلة من أجل اتصال التقارير والبيانات.تتيح لك الوسوم"تنسيق الجدول tableformatting "في الـ HTML(مثل <td>) تنظيم مخرجات جدولية(مستوية) دون الكثير من الجهد. من المؤكد ،أن تنظيم جميع السلاسل النصية المتعددة الحجم مع بعضها لبناء الصفحة هو تنظيم مزعج،ولكن توجد طرق بخصوص ذلك أيضاً.
بالعودة إلى الفصل13،فقد ناقشت فيه XSLT (تحويلات XSL)،طريقة لأخذ بيانات معتمدة علىXML وإعادة شحنها ضمن أي نموذج أخر تريده- متضمناً أعمال عظيمة من الفن،أو HTML المصنوعة بإتقان.مهما يكن فإنك تحصل على HTML،بالإضافة إلى أن لديك خيار عرض الطرق.تتضمن أكثر الطرق المباشرة تخزين HTMLالمنتج في ملف ما للقرص،ويبدأ مستعرض انترنت المستخدم الافتراضي عرضه باستخدام أمر مثل التالي:

```
Process.Start("c:\temp\MyReport.htm")
```

إذا كنت تريد أن يحتوي التقرير مظهر أكثر تكاملاً في تطبيقك،تستطيع عرض محتوى HTMLفي أداة مستعرض ويب web browser .لقد عملنا هذا في كود المشروع العائد للفصل17،عندما عرضنا تفاصيل بند المكتبة كـ HTML.

مستندات XPS Documents .XPS

يتم استخدام أساسيات تقديم النوافذ(WPF)بشكل رئيسي لجعل واجهة المستخدم مكتظة بالألوان والحيوية.ولكن يوجد قسم من التقنية لتوليد مستندات مستقرة معتمدة على XML تُعرَف بـ *XPS*(مواصفات XML الورقيةXML Paper Specification).كما أنك تستطيع إنتاج تقارير باستخدامHTML ،تستطيع إنتاجها باستخدام معيار XPS.يتضمن الفصل18 مناقشة مختصرة عن WPFو XPS.إذا أعجبك إنتاج التقارير باستخدامXPS ،راجع التعليمات والمساعدة المضمنة مع الفيجوال أستوديو.

خدمات وأدوات التقرير.Reporting Services and Controls

تتضمن الفيجوال أستوديو مجموعة من الفئات في فضاء الأسماء Microsoft.Reporting والذي بشكل خاص مصمم من اجل بيانات تقرير report data .الفئة الرئيسية في فضاء الأسماء هذا هي الفئة ReportViewer(تم تعريفها كـ MicrosoftReportViewerفي ذاتية أداتها البديلة).عملياً،إنهما أداتان:واحدة من أجل نماذج ويندوز Windows Forms والأخرى من أجل نماذج الويبWeb Forms .هذه الأدوات تكون معتمدة بشكل جزئي على تقنية موجودة في خدمات التقرير لمخدم ميكروسوفت سكول Microsoft SQL Server Reporting Services،وتستطيع أيضاً استخدام الأداة دون مخدم سكولSQL Server .
سيعمل مشروع المكتبة على استخدام الأداة WinForms.ReportViewer من أجل تقاريره الجاهزة(المبنية داخلياً فيهbuilt-in).سنمضي معظم هذا الفصل في مناقشة هذه الأداة واستخدامها في تطبيقات نماذج ويندوز. ولن أناقش نسخة الأداة الخاصة بنماذج الويب هنا،على الرغم من أنّ استخدامها يوازي استخدام نسخة نماذج ويندوز.
سترى كم هو بسيط إضافة أداة MicrosoftReportViewer إلى مشروع موجود.ولكن الفيجوال أستوديو يتضمن أيضاً قالب مشروع جديد new project template والذي يركز على الأداةMicrosoftReportViewer. يستخدم إنشاء مشروع "تطبيق تقاريرReports Application " جديد المعالج السحري للمشروع project wizard لمساعدتك في تثبيت تقرير خاص.ويمكن أن تكون النتيجة النهائية تطبيق التقرير الكامل،أو تستطيع استخدامه كأساس من أجل تخصيص أبعد(أكثر).

التقارير الصريحة.Crystal Reports

إذا كان لديك على الأقل الطبعة الاحترافية من الفيجوال أستوديو،فسيكون لديك نسخة مجانية من تقارير الكريستال.النسخة المضمنة مجموعة فرعية وظيفية من إصدار الكريستال ريبورتز2008 الرسمي.إذا كنت جديد على الفيجوال بيسك،فقد فاتك النسخة السابقة من الكريستال ريبورتز المضمن مع اللغة منذ إصداراتها الأولية.بسبب هذه العلاقة الطويلة الأمد مع الفيجوال بيسك،فقد أصبحت الكريستال ريبورتز واحدة من حزم التقارير الأوسع استخداماً في السوق.
إن الكريستال ريبورتز مشارك ثانوي،وحالياً مملوك من قبل شركة تدعى " متطلبات العمل Business Objects ".لقد تغيرت الأيدي المالكة لهذا المنتج عدة مرات منذ مزجه لأول مرة مع الفيجوال بيسك، ولكن تبدو شركة "متطلبات العملBusiness Objects "تعتني به الآن.ولن أناقش الكريستال ريبورتز أكثر في هذا الكتاب.

التكامل مع ميكروسوفت أوفيس.Integration with Microsoft Office

لقد أصبحت الفيجوال بيسك لغة الماكرو الرئيسية لمجموعة تطبيقات ميكروسوفت أوفيس المترابطة suiteمنذ موت WordBasic.و لكنني أتحدث عن الفيجوال بيسك السابق للدوت نت،الذي هو غير مدار.لحسن الحظ،تستطيع أيضاً استخدام العالم المدار للفيجوال بيسك2008 للتفاعل مع تطبيقات ميكروسوفت أوفيس.كيفية التفاعل مع أوفيس يعتمد على أياً منهما : مستند أوفيس أو تطبيق الفيجوال بيسك هو المركز الرئيسي للمستخدم.
إذا كان هدفك تحسين تطبيق "مسار العملline of business " باستخدام تطبيقات أوفيس المتنوعة كمدخل portalإلى التطبيق_ على سبيل المثال،إظهار مقدار(أشكال figures) المبيعات الأخيرة من ضمن ميكروسوفت Outlook – خذ بناء تطبيق عمل أوفيس Office Business Application (OBA).يمثل تطبيق عمل أوفيس OBA

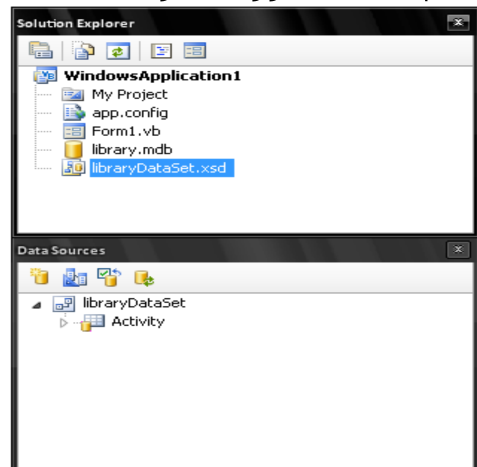
طريقة جديدة في تصميم برامج متكاملة باستخدام الفيجوال أستوديو،ميكروسوفت أوفيسMicrosoft Office ،ميكروسوفت شيربوينت سرفيسس SharePoint Services،والأنظمة الأخرى ذات الصلة.
من ناحية عالم الفيجوال أستوديو،يحدث عمل التطوير من خلال أدوات الفيجوال أستوديو الخاصة(الموجهة)لأوفيسVisual Studio Tools for Office (تختصر كـ VSTO أو visto)المضمنة مع الطبعة الاحترافية وطبعة نظام الفريق للفيجوال أستوديو.
إذا كان هدفك هو إنشاء وظائف إضافية add-ins كـ"شريط مهامtask bar" خاص بك ضمن تطبيقات أوفيس،استخدم قوالب مشروع أوفيس"وظيفة إضافيةAdd-In" الجديدة Add-In project templates المضمنة مع الفيجوال أستوديو.هذه التوسعات السهلة التطوير تتيح لك تخصيص ممارسة(معرفة) المستخدم وذلك بتخصيص مجموعة ميزات أوفيس .تعتبر "الوظائف الإضافيةAdd-ins" أيضاً جزء من فستو visto، وهي متاحة فقط ضمن الفيجوال أستوديو بطبعته الاحترافية أو نظام الفريق.
لو تمكن المستخدم من الوصول إلى ميزات أوفيس بشكل غير مباشر فقط من خلال تطبيق الفيجوال بيسك (على سبيل المثال،إذا كنت تريد من برنامجك أن يبدأ دمج بريد ميكروسوفت ورد)،استخدم مجمعات Interopالرئيسية Primary Interop Assemblies (PIA)لميكروسوفت أوفيس المزودة من قبل ميكروسوفت.توفر هذه المكتبات إمكانية الوصول إلى ميزات معينة لتطبيق أوفيس من خلال فضاء الأسماء Microsoft.Office.مثل VSTO,،هذه المكتبات تربط كود الدوت نت إلى ميكروسوفت أوفيس،ولكن مع التركيز على كودك(كود الدوت نت الذي تكتبه) بدلاً من مستند أوفيس.

استخدام أدوات التقرير في الدوت نت.Using Reporting Controls in .NET

لنمطي ما تبقى من هذا الفصل في مناقشة أدوات التقرير القياسية الموفرة في الفيجوال أستوديو.كما ذكرت سابقاً،يتم تضمين فنتي تقرير ReportViewerفي الفيجوال أستوديو:واحدة من أجل تطوير سطح المكتب وأخرى لتطوير الويب.سأتحدث حول تشكيلة سطح المكتب فقط في هذا الفصل.والمصمم المستخدم لتطوير هذه التقارير لا يفرق بين هدف التقرير(سطح المكتب أو مستعرض الانترنت).توجد بعض الاختلافات في النشر(التوزيع)،ولكن عليّ ترك التوزيع عبر الانترنت كي تبحث عنه أنت.
تتكامل الأداة بشكل مباشر مع خدمات تقرير سكول سرفر،عارضة جميع الصفحات الناتجة بواسطة ذلك النظام المعتمد على المخدم.
بما أننا نفترض أنك تستخدم سكول سرفر بطبعته السريعة من أجل التطوير الخاص بك(والتي لاتحتوي خدمات التقارير).سأركز بالمقابل على تقديم النمط "المحليlocal" للأداة. يتيح لك هذا عرض أي بيانات من أي مصدر تختاره على كل صفحة عرض للتقرير،ومن ضمنها مخدم سكولSQL Server .
لنسلك الخطوات المطلوبة لتصميم تقرير بسيط بشكل مرئي باستخدام الفئة ReportViewer.سنعمل على إنشاء تقرير يعمل على جدولة السجلات في جدول Activityلمشروع المكتبة،الجدول الذي يحتوي بيانات حتى ولو لم تكن قد استخدمت مشروع المكتبة حتى الآن.يعمل هذا بشكل أفضل إذا تتبعت على كمبيوترك مباشرةً،لأن قراءة كيفية تصميم التقرير مشابه لقراءة كيفية إجراء عملية جراحية للمخ:والأكثر منفعةً لو عملت ذلك بشكل عملي.ابدأ بإنشاء تطبيق نماذج ويندوز جديد.
ملاحظة:والآن الأخبار السيئة.
إن الأداة MicrosoftReportViewer ليست الأداة الأسهل للاستخدام في العالم،ولكن الأصعب في استخدامها لو لم تكن قد أنت مع نسختك من الفيجوال أستوديو.إذا كنت تستخدم فيجوال بيسك2008 بطبعته السريعةVisual Basic 2008 Express Edition ،فلن تجد الأداة في صندوق الأدوات.فقد عملت ميكروسوفت على إتاحتها كتحميل منفصل(تمكن من الوصول إلى منطقة التحميل الخاصة بميكروسوفت*http://www.microsoft.com/downloads*،وابحث عن" Microsoft Report Viewer Redistributable 2005 SP1".) ولكن هذا سيمنحك فقط نصف الطريق.سأناقش مصمم فيجوال ريبورتينغ visual reporting designer فيما بعد وهو أيضاً غير موجود في الطبعة السريعة.على الرغم من أنك ما تزال تستطيع إنشاء محتوى XMLوهذا يتم توليده بشكل طبيعي بواسطة المصمم المرئي،فهذا ليس مناسب على الإطلاق.إذا كنت تستخدم النسخة السريعة،ما يزال بإمكانك استخدام كود المشروع في هذا الكتاب. ولن تكون قادر على تصميم تقارير جديدة بشكل مرئي فقطولكن تستطيع تشغيل التقارير المكتوبة مسبقاً والتي ضمنتها مسبقاً،بما أنها مجرد محتوىXML .
بعد كل هذا ،إذا كنت ماتزال تستخدم الطبعة السريعة من الفيجوال أستوديو ،قم بتحميل وتنصيب الملف: Microsoft Report Viewer Redistributable 2005 SP1من موقع ميكروسوفت على الويب.

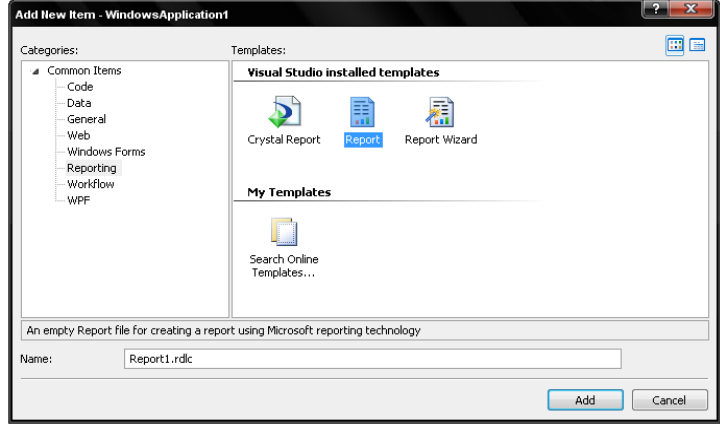
إضافة مصدر البيانات.Adding the Data Source

إضافة مصدر بيانات والذي يشير إلى جدول قاعدة البياناتActivity. فلقد عملنا ذلك سابقاً في الفصل العاشر،في المقطع"إنشاء مصدر بيانات".اختر القائمة: بيانات Data >>إضافة مصدر بيانات جديدAdd New Data Source ، واستخدم المعالج السحري لتركيب مصدر البيانات من أجل البحث عن قاعدة بيانات المكتبةLibrary. عندما تصل إلى قائمة كائنات قاعدة البيانات،ضع علامة صح في الصندوق المجاور لجدولActivity ، وانقر الزر"إنهاء Finish "سيكون لديك الآن مصدر بيانات مسمىLibraryDataSet.
يبين الشكل التالي العناصر المضافة على مستكشف الحلول Solution Explorer ولوحة مصادر البيانات Data Sources بواسطة هذا المقطع.

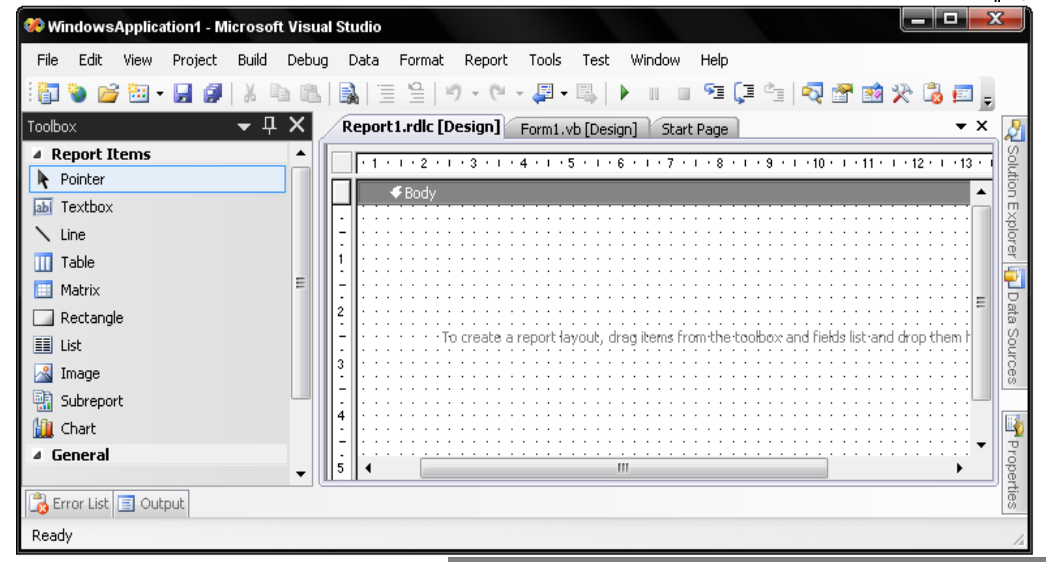


إضافة سطح تصميم تقرير.Adding a Report Design Surface

استخدم القائمة:مشروع Project > > إضافة بند جديد Add New Item لإضافة بند"تقرير"جديد.يبين الشكل التالي بند التقرير في حوار إضافة بند جديد.تأكد من اختيار"تقرير "Report"وليس" Crystal Report" من القائمة.



انقر على الزر إضافة Addلأدراج التقرير ضمن المشروع.يظهر الملف *Report1.rdlc*في مشروعك،ويفتح المصمم الخاص به بشكل آلي.و" RDLC "هو اختصار لـ"لغة تعريف التقرير للعميلReport Definition Language – Client "،وملف هذا النوع يحتوي محتوى XMLوالذي يصف تخطيط التقرير المصمم بشكل محلي.يبين الشكل التالي المصمم الخاص بملف التقرير المضاف،زائد الأدوات في شريط الأدوات toolbarالذي يمكن من إضافة سطح تقرير.سأشير إلى التقارير المنشئة من خلال المصمم كتقارير RDLC على طول باقي هذا الفصل.



تصميم سطح التقرير.Designing the Report Surface

إذا كنت قد كتبت تقارير في ميكروسوفت أكسس أو في أدوات تقرير مشتركة أخرى،من المحتمل أنك على دراية بالتقارير المحزمة banded. هذه التقارير لديها "حزمbands" أو أشرطة stripesتمثل جزء من صفحة الطباعة.تتضمن الحزم رؤوس وتذييل headers and footers،رؤوس وتذييل التقرير،مقطع تفاصيل السجل،ورؤوس المجموعة وتذييلها المستخدمة لتجميع تفاصيل المدخلات بشكل مرئي ومنطقي. عندما يتم تشغيل التقرير ، يبدأ خط أفقي تخيلي على طول عرض الصفحة من أعلى لأسفل الصفحة.وعندما يصطدم الخط بكل حزمة،يعالج التقرير الحقول في تلك الحزمة حتى ينتهي من معالجة جميع السجلات.

تختلف تقارير RDLCقليلاً عن التقارير المحزمة(المقيدة)banded reports .) فيوجد فقط ثلاث أشرطة(حزم)bands :) معنون(رأس)الصفحةpage header ،مذيل(حاشية)الصفحةpage footer، وكل شيء أخر يدعى "الجسمBody ."فبدلاً من إضافة حزم bandsللسجلات records والمجموعاتgroups، تعمل على إضافة حقول fieldsإلى قطاعات البيانات*data regions*. تعالج هذه المتحكمات(أو الأدوات) الخاصة السجلات المربوطة إلى التقارير طبقاً لشكل قطاع البيانات data region. توجد أربعة أدوات لقطاع البيانات data region في صندوق الأدواتtoolbox :

الجدول.Table

يجلب هذا القطاع region عدد غير محدد من صفوف البياناتdata rows،ولكن مع مجموعة معرفة مسبقاً من أعمدة البياناتdata columns. فهو مصمم من أجل التقديم المستوي(الجدولي tabular) لسجلات البيانات،في كل عمود يتم عرض مصدر مفرد أو حقل بيانات محسوب بشكل عام.كل صف من الجدول يمثل سجل بيانات المصدر.

القالب.Matrix

تشابه هذه الأداة حقل الجدولTable، ولكنها تسمح بعدد مرن من أعمدة البيانات،وليس فقط الصفوف.

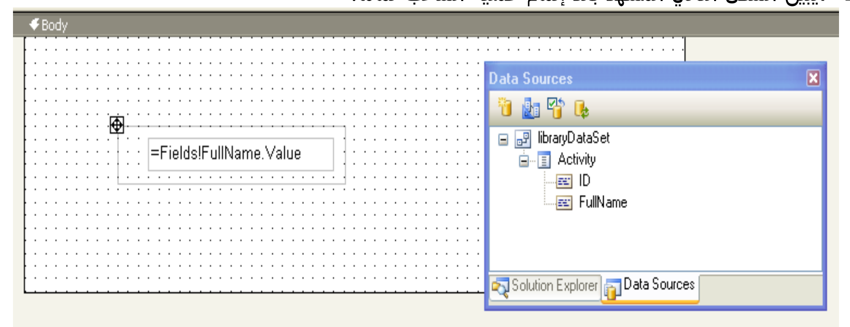
القائمة.List

يوفر قطاع القائمة List region مقطع عرض حر النموذج free-form display section لكل سجل قادمincoming record. تستطيع إضافة أي عدد من الحقول أو أدوات عرض display controls إلى مقطع السجلrecord section .

المخطط.Chart

تستخدم المخططات Chartsالبيانات المحسوبة(المجمعة)من التقرير لإحضار خط،شريط bar ،ومخططات دائرة القطاعات النسبية(الدائرة المقسمة كنسب مئوية pie charts) إلى المستخدم.

يتم دائماً ربط التقارير من مجموعات بيانات إلى قطاع بيانات data region. إذا كان تقريرك يتضمن بيانات من مصادر بيانات متعددة مختلفة،فسيرتبط كل مصدر بيانات إلى قطاع region تقرير واحد بالضبط، وتظهر جميع الحقول(التقسيمات) في الحزمة Body.سنستخدم قطاع البيانات"قائمة List " من أجل عينة التقرير هذه. استمر الآن وأعمل على إضافة أداة "قائمة List " إلى الحزمة"جسم Body " على سطح التقرير.تستطيع الآن إضافة بنود أخرى إما إلى سطح الحزمة نفسها،أو إلى سطح الأداة"قائمةList ."البنود المضافة إلى أداة "القائمة List "يتم إعادة معالجتها من أجل كل سجل في مصدر البيانات القادم.يمكن لهذه البنود إما أن تكون أدوات controlsمن صندوق الأدوات toolbox أو حقول قاعدة بيانات database fields معروضة في لوحة مصدر البيانات Data Sources panel. استخدم الجدول Activityفي لوحة مصدر البيانات،اسحب حقل FullNameإلى سطح الأداة"قائمة List "،يبين الشكل التالي المشهد بعد إتمام عملية السحب تماماً.



عندما سحبنا الحقل من مصدر البيانات إلى أداة"القائمة List "عملت الفيجوال أستوديو على تأسيس اتصال بينهم.يشير حقل DataSetName لأداة القائمة list1الآن إلى LibraryDataSet_Activity،وهو اسم مصدر البيانات. وأضافت الفيجوال بيسك أيضاً أداة صندوق نص TextBoxإلى سطح أداة القائمة،وأضافت التعبير(=Fields!FullName.Value)) الذي يعرض محتوى ذلك الحقل من قاعدة البيانات من أجل كل سجل معالج.

سأعمل على إعادة تحجيم أداة القائمةList ،صندوق النصtext box ،وحزمة الجسم Bodyبحيث يشغل حقل صندوق نص FullNameجميع سطح التقرير.كما هو مبين في الشكل التالي.

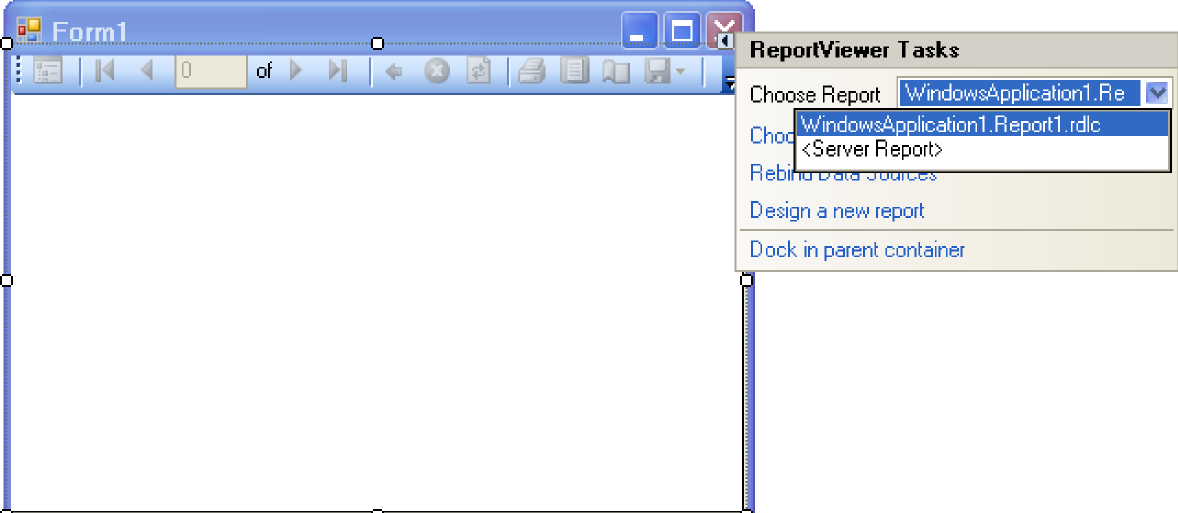
=Fields!FullName.Value

التقرير جاهز الآن للاستخدام.عندما نصمم سطح التقرير،تكون الفيجوال أستوديو مشغولة في توليد XMLوتخزينه في الملف*Report1.rdlc* .

استخدام أداة التقرير.Using a Report Control

إن ملف RDLCهو مجرد تعريف XMLلتقرير،ليس لديه القدرة على عرض نفسه.لعرض التقرير،يجب علينا إضافة أداة تقرير إلى الفورم أو صفحة ويب التي تعرف كيف تمزج محتوى تصميم XMLمع البيانات dataبشكل مناسب من مصدر بيانات معين.ارجع إلى الفورم Form1وأضف الأداة MicrosoftReportViewerإلى سطح هذه الفورم من صندوق الأدوات(وهذه الأداة موجودة في مقطع التقارير Reporting لصندوق الأدواتtoolbox .)تتضمن الأداة المضافة زر وسوم الاستشعار الصغير" small “smart tags” button في الزاوية العلوية اليمينية".إن نقر هذا الزر يعرض نافذة سريعة بمهمات ReportViewer،والتي تظهر في الشكل التالي.

تجلب الأداة MicrosoftReportViewerتجربة معتمدة على فورم لعرض التقارير.معظم الأداة عبارة عن مساحة فارغة حيث سيظهر التقرير.وهي تتضمن أيضاً شريط أدوات toolbarيستخدم للتنقل navigateخلال صفحات التقرير.يستطيع المستخدم أيضاً البدء بإخراج طبعة للتقرير من خلال هذه الأدواتcontrols. إذا كنت لاتحتاج شريط أدوات أو واحد من أدواته،استخدم خاصية التشكيلة ...Showلأداة MicrosoftReportViewerلإخفاء البنود الغير مطلوبة.



عارض التقرير report viewer شامل وغير معتمد على التقرير.إذا كان لديك عدة ملفات RDLC في مشروعك،تستطيع عرض أياً منها(واحد كل مرة)من خلال نفس عارض التقرير.لدينا الآن تقرير واحد في مشروعنا،لذلك دعنا نصله(*Report1.rdlc*)إلى العارض باستخدام المهمة "اختر تقرير Choose Report "كما هو مبين في الشكل السابق

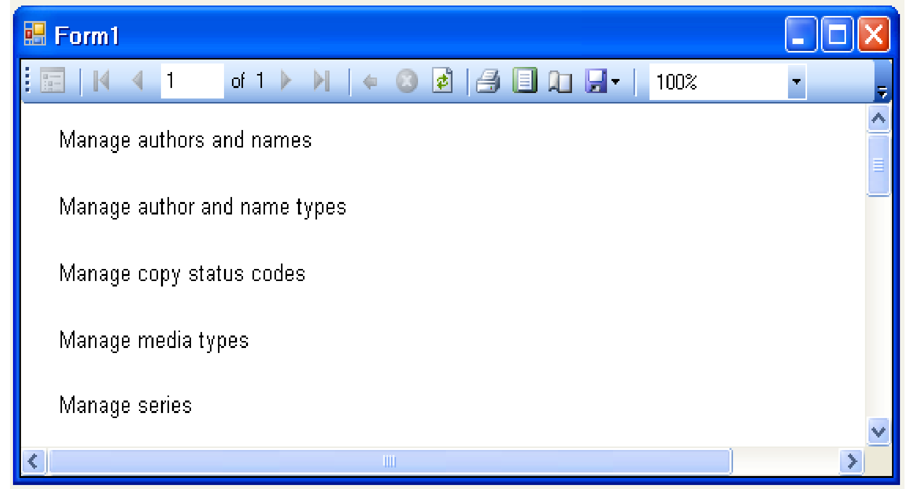
وذلك من report viewerزر وسم الاستشعار smart tag button.أنقر أيضاً"ترصيف في الحاوية الرئيسية Dock in parent container " في النافذة السريعة لتمديد التقرير على حجم الفورم.

إن كل من تقرير RDLCوالبيانات dataمن مصدر البيانات data source والأداة MicrosoftReportViewer جميعها تنضم في عرض تقرير رائع بواسطة سحر تحزيم البيانات data binding. عندما تربط التقرير بأداة العرض،تظهر ثلاث أدوات إضافية على الفورم: LibraryDataSet، ActivityBindingSource، و LibraryDataSetوهي مرجع إلى مصدر البيانات الفعلي الذي أضفناه سابقاً.أما الأداتين المتبقيتين تعملان على تضمين البيانات في الفورم بحيث يمكن تحزيمها إلى عارض التقرير. على الرغم من أنك لا تستطيع رؤيتها في المصمم،كود الفورم المخفي يعمل على وصل هذه الأدوات وتقرير XML إلى العارض.

```
Me.ReportViewer1.Dock = System.Windows.Forms.DockStyle.Fill
        ReportDataSource1.Name = "libraryDataSet_Activity"
        ReportDataSource1.Value = Me.ActivityBindingSource
        Me.ReportViewer1.LocalReport.DataSources.Add(ReportDataSource1)
        Me.ReportViewer1.LocalReport.ReportEmbeddedResource = "report.Report1.rdlc"
        Me.ReportViewer1.Location = New System.Drawing.Point(0, 0)
        Me.ReportViewer1.Name = "ReportViewer1"
        Me.ReportViewer1.Size = New System.Drawing.Size(401, 246)
        Me.ReportViewer1.TabIndex = 0
```

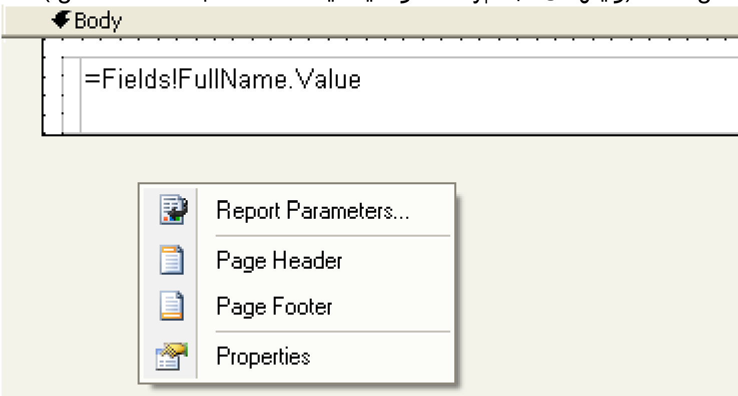
تشغيل التقرير.Running the Report

اضغط F5وشاهد نتيجة جهودك.يبين الشكل التالي النتيجة.لقد عملت على ضبط العرض بالنقر على زر شريط الأدوات"تخطيط الصفحةPage Layout "،ووضعت مستوى التكبير zoom level إلى عرض الصفحةPage Width. كما ترى إن التقرير مناسب،ولكن نستطيع جعله أكثر أناقةً .



إضافة رأس وتذييل صفحة.Adding a Page Header and Footer

أظن أن التقرير يحتاج إلى عنوان ذو معنى عند أعلى كل صفحة،زائد رقم الصفحة في الزاوية السفلية اليمينية.دعنا نعود على مصمم التقرير RDLC ونعمل على إضافتهم.حالما تكون هناك،انقر يمين على خلفية التقرير background (وليس عل الجسمbody، والذي عليه علامات الشبكةgrid marks)،كما هو مبين في الشكل التالي.



من هذه القائمة اختر"رأس الصفحة(أو معنون الصفحةPage Header)"ومن ثم أحضر هذه القائمة مرةً أخرى واختر"مذيل الصفحةPage Footer ".كل حزمة جديدة تظهر على سطح التقرير.

فيما إذا كانت هذه الحزم الجديدة مستقرة،نص غير متغير أو نص يتم توليده بشكل حركي dynamicallyمن مصدر بيانات.إن أداة صندوق النص TextBoxهي الأداة الاختيارية لإظهار محتوى نص. أضف أداة صندوق نص TextBoxمن صندوق الأدوات toolboxإلى كلا المقطعين"رأس ومذيل الصفحة" .انقر داخل صندوق نص معنون الصفحة واكتب التالي:

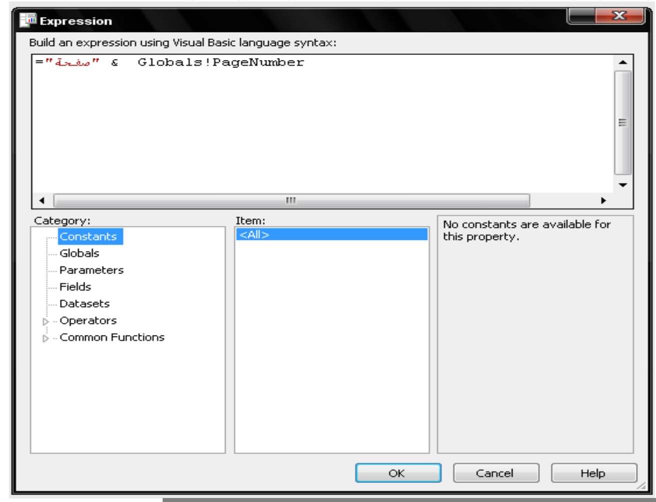
="تقرير عن جدول النشاط"

تستطيع استخدام لوحة الخاصيات لضبط مظهر هذه الأداة،متضمنة خط العرض.

في صندوق نص المذيل اكتب هذا النص:

="صفحة" & Globals!PageNumber

يتضمن شبه الكائن Globalsعدة أعضاء تستطيع استخدامها في التقرير.ولكن كيف عرفت استخدام: "صفحة" Globals!PageNumper &؟لقد عملت على بناء التعبير بشكل مرئي باستخدام محرر التعبيرExpression Editor .للوصول له ،انقر يمين على أداة صندوق النص TextBoxواختر تعبير Expression من القائمة المختصرة.يبين الشكل التالي هذا المحرر،فهو يتيح لك بناء تركيب باستخدام قوائم من الدوال وأسماء الحقول.يحدث لأن تكون الدوال الفعلية نداء(حث) لدوال الفيجوال بيسك.



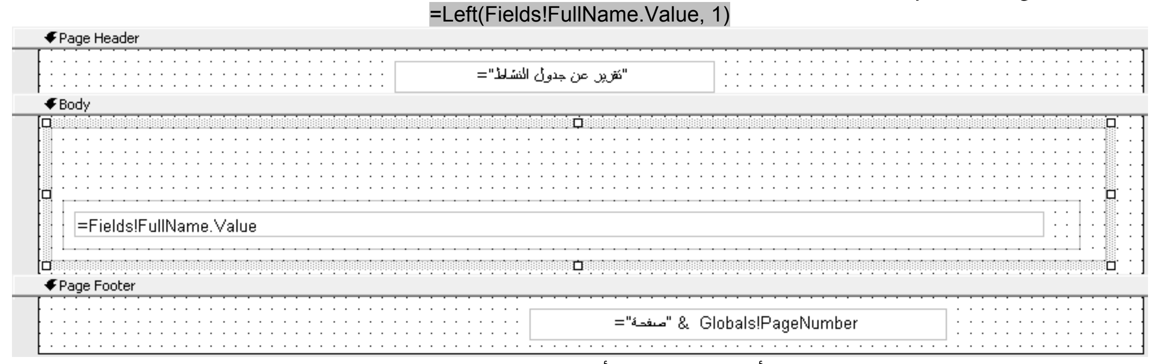
دعم التجميع والترتيب.Support for Grouping and Sorting

تجميع البيانات شائع في التقارير المطبوعة.لإضافة تجميع لتقريرنا،نحتاج إلى تضمين أداة القائمة Listالموجودة(سجل التفاصيلthe detail record) ضمن أداة قائمة List أخرى(المجموعةthe group)،ووضع الخاصيات المتنوعة على أداة قائمة المجموعة group List لتحديد طريقة تجميع البيانات.

لنعمل على إضافة أداة قائمة أخرى(مسماة list2)إلى جسم التقرير،وامنحها ضعف ارتفاع أداة القائمة الموجودة(المسماة list1)ومن ثم،اسحب list1(سجل التفاصيل detail record)لضمن list2(المجموعة الجديدةthe new group)،ضعها باتجاه الأسفل.سيبدو تقريرك كما في الشكل التالي.

لتركيب المجموعة،انقر يمين عليها واختر الخاصيات Propertiesمن القائمة المنسدلة.يظهر نموذج خاصيات القائمةList .على تبويبها"عام General "،انقر الزر"تحرير مجموعة التفاصيلEdit details group "،والذي يضع التجميعgrouping .على الخاصيات "ترتيب Sorting " و"تجميع Grouping "التي تظهر، أدخل النص التالي ضمن الصف الأول first row لحقل"تجميع على Group on ":

=Left(Fields!FullName.Value, 1)



هذا التعبير يخبر أداة list2تجميع نتائج تفاصيلها بواسطة الحرف الأول لحقل الاسم الأول.

على نفس هذه الفورم،أضف النص التالي إلى الحقل" Document map label ":

="Letter: " & Left(Fields!FullName.Value, 1)

إن "تشبيك مستند document map "يمكِّن قائمة رابطة فائقة(رابطة مدمجةhyperlink) ضمن المجموعات المختلفة للتقرير.عندما نشغل التقرير بعد قليل،سنرى هذا الربط إلى اليسار تماماً على سطح عرض التقرير.

السجلات في جدول Activityيتم ترتيبها من أجل الراحة بالنسبة للمبرمج.ولكن من المحتمل أن يطلب مستخدم التقرير رؤيتها مرتبة بزي مقبول نوعاً ما.انقر على التبويب"ترتيب Sorting "،وأضف النص التالي إلى الحقل"ترتيب على Sort on "في عمود التعبيرExpression :

=Fields!FullName.Value

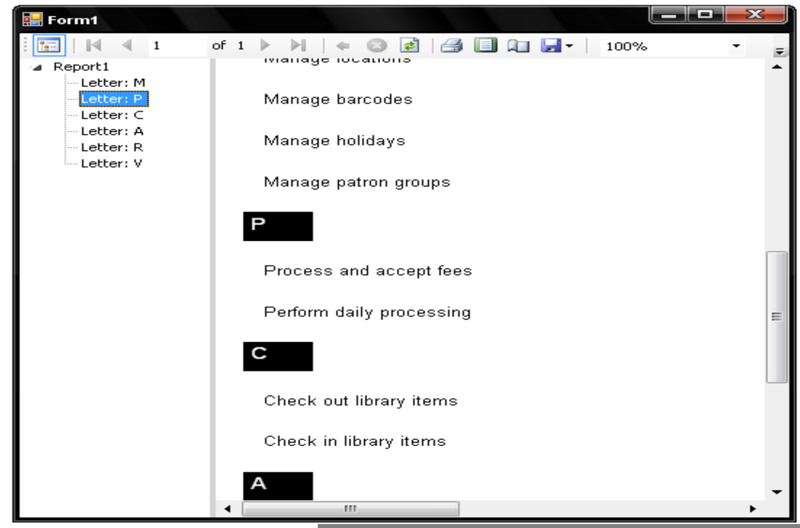
كما تتوقع،هذا سيرتب البيانات بواسطة حقلFullName . انقر الزر "موافق OK "وارجع إلى سطح التقرير.

سنبقى بحاجة لإضافة شيء ما يجعل كل مجموعة أفضل بشكل ملحوظ. أضف أداة صندوق نص TextBox إلى أداة التجميعlist2 . ضعها في الزاوية العلوية اليسارية للأداة الأم،واكتب النص التالي ضمنها(أو ضمن خاصيتها Value):

=Left(Fields!FullName.Value, 1)

ولقد عملت أيضاً على وضع خاصية BackgroundColorإلى"أسودBlack "،وخاصية "اللون Color "إلى أبيضWhite، وخاصية "الخط Font "إلى"عاديNormal،إلىArial،نقطة 12pt،غامقBold "من أجل المظهر فقط .

تشغيل التقرير يمنحك النتائج المبينة في الشكل التالي.لاحظ ربط المستند على طول الحافة اليسارية للنافذة،وعناوين الحرف المفرد المجمعة قبل كل مقطع مجموعة.



تنسيق التخطيط المحسن.Enhanced Style Formatting

من المحتمل أن الميزة الأسهل لتقارير RDLCهي أنه يمكن للعديد من الخاصيات لبند تم وضعه على سطح التقرير تضمين تعابير شرطية.وهذا يعني أن بإمكانك تعديلها بشكل شرطي،يعني،أن الخاصيات المرئية لأداة صندوق النص TextBoxتعتمد على قيمة حقل في السجل الحالي.
في مقطع مشروع هذا الفصل،سنكتب تقرير يستخدم تواريخ استحقاق الدفع due dates للبنود المستخرجة حالياً.إذا كان البند مستحق الدفع سابقاً،فأريد إظهار تاريخ الاستحقاق بالأحمر.عادة خاصية اللون Colorلأداة صندوق النص TextBox (التي هي لون خط الأدوات)هي أسودBlack .،لأجعل ذلك الحقل يستجيب لبنود مستحقة الدفع،سأعمل على تبديل"الأسود"بالتعبير التالي:

=IIf(Fields!DueDate.Value < Today, "Red", "Black")

استخدام بيانات خاصة.Using Custom Data

على الرغم من أنه شائع كثيراً إنتاج التقارير من قاعدة البيانات،تستطيع عملياً استخدام بيانات من أي مصدر.عند استخدام الأداةMicrosoftReportViewer، فأي مصدر بيانات ينفذ واجهة IEnumerable هو جيد بشكل كافي. وهذا يتضمن جميع التجمعاتcollections، المصفوفاتarrays، ونتائج استعلام لينكوLINQ. فالتقرير ليس بتلك الانتقائية(الحرص picky)،طالما أنه تم تنسيق البيانات كما هو متوقع.من أجل التقرير الذي عملناه، بإمكاننا التخلص من ditchالبيانات الفعلية وتوفير بيانات مزيفة خاصة بنا.يجب علينا إتباع القليل من القواعد لجعله يعمل:

- عندما سحبنا الحقل Activity.FullNameمن مصدر البيانات إلى سطح التقرير، حصل التقرير(عملياً،أداة القائمة list1) على الفكرة البسيطة التالية أن جميع البيانات يجب أن تأتي من مصدر بيانات مسمى LibraryDataSet_Activity.وأي مصدر بيانات نستخدمه مكان البيانات الحقيقية يجب أن يحتفظ بهذا الاسم.

- مصدر البيانات المزيف يجب أن يتضمن حقلFullName، بما أنه هو الحقل الذي يتوقعه التقرير.

هذه القواعد ليست بهذا السوء.لذلك إليك ما نحتاج إليه من أجل العمل:إنشاء مصدر بيانات مزيفfake data source، اعتراض سبيل interceptالتقرير تماماً قبل محاولته الحصول على البيانات من قاعدة بيانات المكتبة،وإدخال بياناتنا الخاصة عوضاً عنه.
من أجل مصدر البيانات المزيف،سنحتاج إلى فئة تتضمن على الأقل الحقل FullName.

```
Public Class FakeActivityRecord
    Private StoredID As Long
    Private StoredFullName As String
    Public Sub New(ByVal whatID As Long,ByVal whatFullName As String)
        StoredID = whatID
        StoredFullName = whatFullName
    End Sub
    Public Property ID() As Long
        Get
            Return StoredID
        End Get
        Set(ByVal value As Long)
            StoredID = value
        End Set
    End Property
    Public Property FullName() As String
        Get
            Return StoredFullName
        End Get
        Set(ByVal value As String)
            StoredFullName = value
        End Set
    End Property
End Class
```

الحقول المعروضة يجب أن تكون خاصيات،وليست مجرد حقول(متغيرات)عامة،فعارض التقرير لا يميز حقول الأعضاء القياسية.

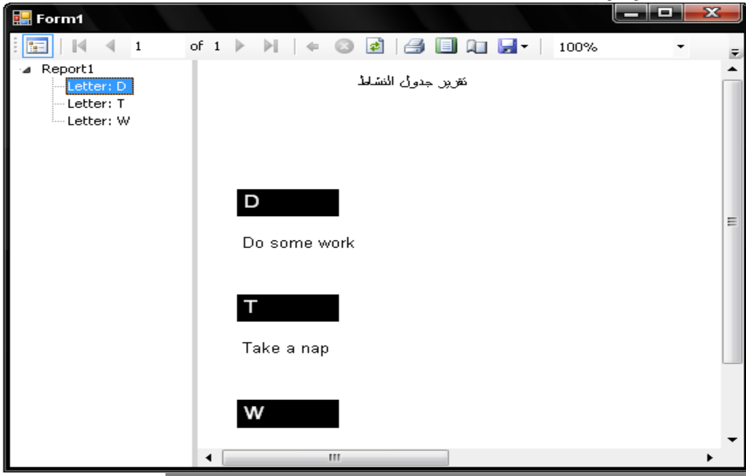
إذا ألقيت نظرة على الكود المصدري لـ Form1،ستجد الكود التالي وقد تم إضافته إلى معالج حدث تحميل الفورم Form_Load عندما عملنا على وصل مستعرض التقرير مع تقرير RDLC.

```
Me.ActivityTableAdapter.Fill(Me.libraryDataSet.Activity)
        Me.ReportViewer1.RefreshReport()
```

حيث أن السطر الأول يحمل البيانات من جدول قاعدة بيانات المكتبة ويعمل على وصلها بالتقرير في السطر الثاني.نحتاج إلى استبدال هذه الأسطر المولدة من قبل المعالج السحري بكود يقطع الطريق على اتصال البيانات الحقيقة.

```
' ----- Create a fake table of fake records.
    Dim fakeSource As New Collections.Generic.List(Of FakeActivityRecord)
    ' ----- Add each of the fake records.
fakeSource.Add(New FakeActivityRecord(1, "Do some work"))
fakeSource.Add(New FakeActivityRecord(2, "Take a nap"))
fakeSource.Add(New FakeActivityRecord(3, "Write a program"))
    ' ----- The report was already bound to the true
    ' data source. Delete it.
Me.ReportViewer1.LocalReport.DataSources.Clear( )
    ' ----- Build a new data source. Remember, it must have
    ' the same name.
    Dim fakeReportSource As New Microsoft.Reporting.WinForms.ReportDataSource
fakeReportSource.Name = "LibraryDataSet_Activity"
fakeReportSource.Value = fakeSource
    ' ----- Connect the data source to the report, and we're done.
Me.ReportViewer1.LocalReport.DataSources.Add(fakeReportSource)
Me.ReportViewer1.RefreshReport( )
```

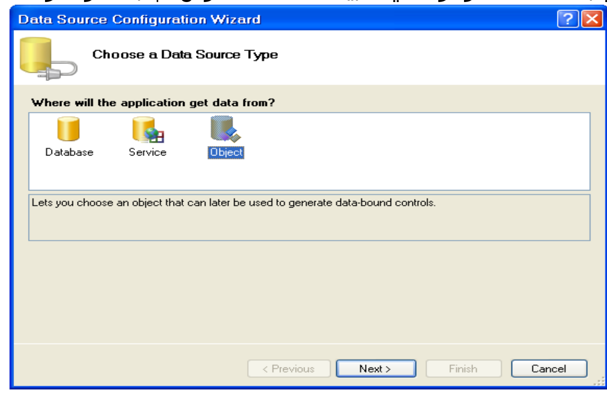
يبين الشكل التالي التقرير مع البيانات المزيفة على المستعرض.



توفير مصدر بيانات خاص.Supplying Custom Data Sources

إن استبدال البيانات في المقطع السابق جيد،ولكن إذا كنت تريد تصميم تقرير لايعتمد على قاعدة بيانات على الإطلاق،تستطيع عمل ذلك،أيضاً،وذلك بتوفير مصدر بيانات خاص بالكامل.تحتاج تقارير RDLCنوع ما من تخطيط مصدر البيانات وقت التصميم، ما لا تستطيع عمله فقط هو توفير بيانات خاصة بالكامل أثناء التشغيل عند تشغيل التقرير. ولكن تستطيع توفير تخطيط خاص معتمد على فئة في تطبيقك.من أجل الفئة، سنرتبط بالفئة FakeActivityRecordالتي عملنا على إنشاءها منذ قليل.ومن ثم سنصمم مصدر بيانات من هذه الفئة.اختر القائمة بيانات Data<< Add New Data Source إضافة مصدر بيانات جديد.عندما يظهر المعالج السحري لتركيب مصدر البيانات ،ففي الماضي كنت دائماً تختار قاعدة بيانات Databaseكمصدر للبيانات.في هذه المرة،اختر "كائنObject " كما هو مبين في الشكل التالي.

عندما تنقر على زر"التالي Next"،يظهر تنظيم هرمي بكل الفئات الموجودة في تطبيقك.مدد الفئات، ومن ثم ابحث واختر الفئة FakeActivityRecord.



انقر على الزر إنهاء.تظهر الفئة FakeActivityRecordكمصدر بيانات في لوحة مصدر البيانات.
تستطيع الآن سحب وإسقاط الحقل FullNameلمصدر البيانات على سطح تصميمي لتقرير RDLC جديد.أضف تقرير جديد لمشروعك،واتبع نفس الخطوات التي استخدمناها سابقاً لتصميم التقرير الأول.هذه المرة،استخدم مصدر البيانات FakeActivityRecord بدل مصدر البيانات LibraryDataSet.لاختبار هذا التقرير الجديد ،يمكنك أنشاء مشروع جديد وإضافة الأداة MicrosoftReportViewer إلى الفورم وترصيفها على سطح الفورم.ولكن لا تعمل على ربطها بالتقريرRDLC. وهذا يجعل الأشياء أوضح بما انه لايوجد أدوات تحزيم مصدر و أي شيء أخر. ومن ثم اعمل على إضافة الكود التالي إلى معالج حدث تحميل Loadالفورم.(ولكن لا تنسى إضافة الفئة FakeActivityRecordإلى المشروع الجديد).كما سترى فقد علمت على تسمية المشروع الجديد" reportcostomdatasource "والتقرير" Report1"

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        ' ----- Link to the RDLC report design.
        Me.ReportViewer1.LocalReport.ReportEmbeddedResource = "reportcostomdatasource.Report1.rdlc"
        ' ----- Create a fake table of fake records.
        Dim fakeSource As New Collections.Generic.List( Of FakeActivityRecord)
        ' ----- Add each of the fake records.
        fakeSource.Add(New FakeActivityRecord(1, "Breakfast"))
        fakeSource.Add(New FakeActivityRecord(2, "Lunch"))
        fakeSource.Add(New FakeActivityRecord(3, "Dinner"))
        ' ----- Build a new data source. Remember, it must have
        ' the same name.
        Dim fakeReportSource As New Microsoft.Reporting.WinForms.ReportDataSource
        fakeReportSource.Name = "reportcostomdatasource_FakeActivityRecord"
        fakeReportSource.Value = fakeSource
        ' ----- Connect the data source to the report, and we're done.
        Me.ReportViewer1.LocalReport.DataSources.Add(fakeReportSource)
        Me.ReportViewer1.RefreshReport()
        Me.ReportViewer1.RefreshReport()
    End Sub
```

إنه مشابه كثيراً للكود الخاص السابق،على الرغم من أن مصدر البيانات الآن هو reportcostomdatasource_FakeActivityRecord ،لقد علمت على حفظ نسخة من عمل هذه التقارير ،اذهب إلى دليل المشروع للفصل21 وراجع هذه التقارير.

Project.مشروع

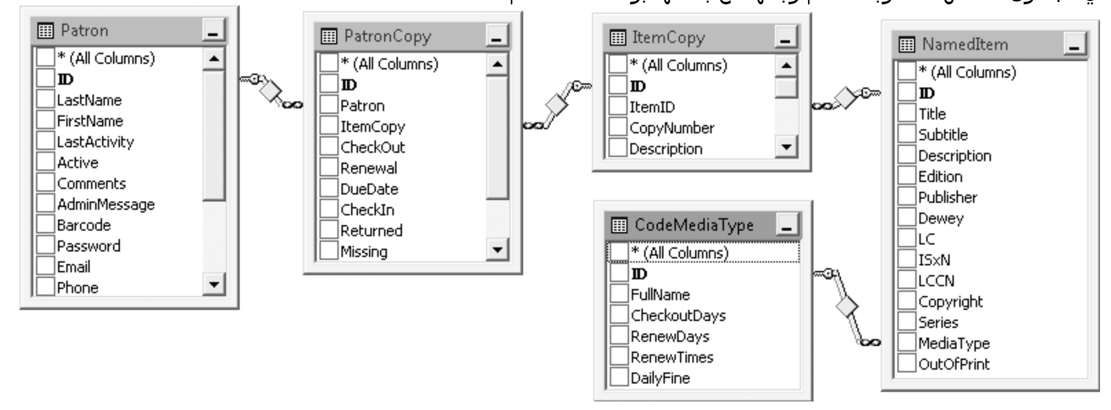
عندما غادرنا مشروع المكتبة للمرة الأخيرة تركنا مستند مجموعة الموارد التقنية،وهي تحتوي على خمس تقارير جاهزة:

التقرير#1:تقرير البنود المخرجة.Items Checked Out Report.
التقرير#2:تقرير البنود المتأخرة الإعادة.Items Overdue Report.
التقرير#3:تقرير البنود المفقودة.Items Missing Report.
التقرير#4:تقرير الغرامات المستحقة على الزبائن.Fines Owed by Patrons Report.
التقرير#5:تقرير إحصاءات قاعدة بيانات المكتبة.Library Database Statistics Report.

سنعمل على إضافة هذه التقارير الخمسة إلى المشروع في هذا الفصل.قبل كتابة أي كود،نحتاج إلى استكشاف كيفية الحصول على البيانات.بما أن البيانات ستأتي من قاعدة بيانات المكتبة،ما نحتاج إليه صناعة عبارة سكول من أجل كل تقرير سنعمل على ربطه إلى التقرير المصمم.
التقرير الخامس"الإحصاءات" سيخبر عن أشياء مثل عدد البنود،عدد الزبائن،وقيم إحصائية أخرى من قاعدة بيانات المكتبة.بما أن هذه البيانات تأتي حقيقةً من عبارة سكول واحدة، سنعمل على استخراج البيانات من قاعدة البيانات ونبني مورد بيانات خاص يغذي التقرير.

Crafting the SQL Statements.نحت عبارة سكول

يعمل التقرير الأول"البنود المخرجة items checked out" ، على جدولة اسم الزبون وعنوان البند من أجل كل بند قيد الإخراج من قبل زبون.يتضمن الجدول"الزبون Patron "(من أجل الحصول على اسم الزبون)، والجدول"نسخة ا لزبون PatronCopy "(حدث الإخراجcheckout) ،والجدول" نسخة بند ItemCopy "(البند الفعلي الذي تم إخراجه)،والجدول"البند المسمى NamedItem "(حيث يظهر عنوان البند)،سنضمن أيضاً الجدول" نوع كود الوسيطة CodeMediaType "،والذي يخبرني فيما إذا كان البند كتاب أو سيدي CD، أو نوع وسيطة أخرى.
تتضمن منصة إدارة ميكروسوفت سكول سرفر السريع Microsoft SQL Server Management Studio Expressمصمم استعلام مرئي نستطيع استخدامه لتصميم الاستعلام.يبين الشكل التالي الجداول الخمس المطلوبة كما تم ربطها مع بعضها بواسطة المصمم.



فيما إذا كنت تستخدم مصمم الاستعلام أو تبني عبارات سكول يدوياً،ستنتهي بشيء مشابه للتالي،وهذا ما سنستخدمه في تطبيق المكتبة:

```
/* ----- Report #1: Items checked out report. */
SELECT     PA.LastName + ', ' + PA.FirstName AS PatronName, PA.Barcode AS PatronBarcode,
PC.DueDate, IC.CopyNumber, IC.Barcode AS ItemBarcode, NI.Title ،
                      CMT.FullName AS MediaName
FROM         Patron AS PA INNER JOIN
                      PatronCopy AS PC ON PA.ID = PC.Patron INNER JOIN
                      ItemCopy AS IC ON PC.ItemCopy = IC.ID INNER JOIN
                      NamedItem AS NI ON IC.ItemID = NI.ID INNER JOIN
                      CodeMediaType AS CMT ON NI.MediaType = CMT.ID
WHERE     (PC.Returned = 0) AND (PC.Missing = 0) AND (IC.Missing = 0(
ORDER BY NI.Title, IC.CopyNumber, PA.LastName, PA.FirstName
```

يربط هذا الاستعلام جميع الجداول،ومن ثم يستعلم عن كل سجل لم يعمل على إعادة قيمة(PC.Returned = 0).ويتجاهل أي بند تم تعليمه"مفقود")(PC.Missing = 0 AND IC.Missing = 0).وهذا الاستعلام سيقود التقرير في النهاية..ولكن حالياً،اعلم أن التقارير RDLCلاتحتاج إلى عبارة سكول أو جدول قاعدة بيانات من أجل مخطط التقرير.نستطيع أيضاً بناء مخطط ملائم يدوياً باستخدام فئة.ويبدو هذا أوضح بما أننا لانريد أن يكون لدينا الكثير من الحقول المرتبطة بمجموعة البيانات المنثورة على طول الكود المصدري للمشروع.(مورد البيانات LibraryDataSetالذي عملنا على إنشاءه في مثال التقرير السابق عمل على إضافة أربع ملفات مصدرية وحوالي 50كيلوبايت من الكود المصدري للمشروع،بدون احتساب التقرير،أما مصدر البيانات المعتمد على فئة لايعمل على إضافة أي كود ما عدا تعريف الفئة نفسها والقليل من XMLفي ملف RDLC.)

أما بالنسبة لمخطط مورد البيانات،نستطيع استنتاجه من الشرط SELECTلاستعلام سكول.إذا كان علينا تصميم فئة مع التخطيط الموافق،سيبدو كما يلي(بدون كود تفاصيل الخاصيات).

```
Class Report1Schema
Public Property PatronName() As String
Public Property PatronBarcode As String
Public Property DueDate As Date
Public Property CopyNumber As Integer
Public Property ItemBarcode As String
Public Property Title As String
Public Property MediaName As String
End Class
```

التقريرين التاليين من " البنود المتأخرة الإستعادة" و"البنود المفقودة" بالنسبة لي،مخطط التقرير الأول هو ما تريد رؤيته بالضبط في التقريرين التاليين،لذلك سنستخدم نفس عبارة سكول تماماً.ما نحتاج عمله هو تغير الشرط WHERE.من أجل تقرير البنود المتأخرة الإعادة،استخدم شرط WHERE التالي:

```
WHERE PC.Returned = 0
AND PC.Missing = 0
AND IC.Missing = 0
AND PC.DueDate < GETDATE( )
```

سيستخدم تقرير البنود المفقودة شرط WHEREالتالي:

```
WHERE PC.Missing = 1
OR IC.Missing = 1
```

يعر التقرير الرابع كمية الغرامات المتبقية المستحقة على زبون،لذلك سيتطلب منا تخطيط مختلف.إليك عبارة سكول،والتي تستخدم بعض ميزات تجميع الإجمالي.

```
----- */Report #4: Fines owed by patron/* .
SELECT PA.LastName + ', ' + PA.FirstName AS PatronName،
PA.Barcode AS PatronBarcode،
SUM(PC.Fine - PC.Paid) AS FinesDue
FROM Patron AS PA
INNER JOIN PatronCopy AS PC ON PA.ID = PC.Patron
GROUP BY PA.LastName + ', ' + PA.FirstName, PA.Barcode
HAVING SUM(PC.Fine - PC.Paid) > 0
ORDER BY PatronName
```

إليك التخطيط الذي يوافق التقرير الرابع:

```
Class Report4Schema
Public Property PatronName() As String
Public Property PatronBarcode As String
Public Property FinesDue As Decimal
End Class
```

أما من أجل التقرير الأخير،سنستخدم فقط تخطيط مع قيمتين من نوع السلسلة النصية:اسم الاحصائية،والقيمة المتعلقة بها.إليك تخطيطها:

```
Class Report5Schema
Public Property EntryName() As String
Public Property EntryValue As String
```

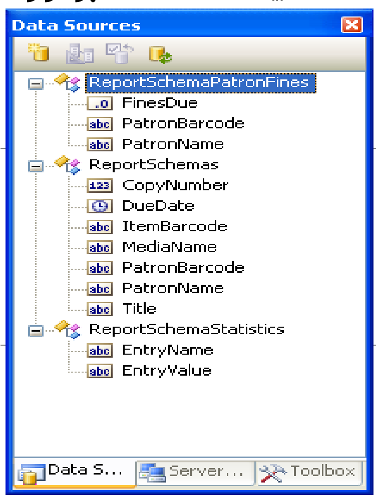
```
End Class
```

حسناً،يكفي تحضير،لندخل إلى المشروع ونبدأ بكتابة الكود .

إضافة تخطيطات التقرير.Adding Report Schemas

عملت على إضافة الملف ReportSchemas.vb إلى المشروع،وهو يتضمن المخططات الثلاث المستخدمة من أجل التقارير الخمسة المبنية داخلياً.راجع المشروع لتلقي نظرة على هذه الملف .

ما تحتاج إليه الآن بعد بناء مخطط الفئات في المشروع بناء المشروع قبل أن تتمكن من استخدام هذه الفئات في تقارير RDLCكمصادر بيانات .في مشروع المكتبة،اعمل على بناء المشروع الآن من خلال القائمة بناء Build>> الأمر بناء المكتبة Build Library.جميع هذه المخططات يجب أن تظهر عندئذٍ كموارد في لوحة موارد البيانات(شاهد الشكل التالي).إذا كانت لوحة موارد البيانات مغلقة اعمل على فتحها من خلال القائمة بيانات Data>> أظهر موارد البيانات Show Data Sources.



إضافة تقارير.Adding Reports

سأعمل هنا على بناء خمس تقارير RDLCجاهزة :

ReportCheckedOut.rdlc

ينفذ هذا الملف التقرير رقم1،تقرير"البنود المخرجة أو المستعارة".وهو يستخدم تخطيط الفئة ReportSchemaPatronItems،ويتضمن ثلاث أعمدة في قائمة البيانات الرئيسية:اسم الزبون/كود التعريف،اسم البند/البار كود/التفاصيل،وتاريخ استحقاق الإعادة. من أجل حقل اسم البند،أردت إحضار معلومات إضافية عندما تكون متاحة. إن كل من اسم البند،رقم النسخة،ونوع الميديا قيم مطلوبة،ولكن كود تعريف بند هو اختياري،إليك التنسيق الذي أفضله:

اسم البند(#رقم النسخة،نوع الوسيطة،كود التعريف).Item Name (#CopyNumber, MediaType, Barcode)

للحصول على النتيجة،كان عليّ جمع حقول الموارد المتنوعة مع بعضها،واستخدام الدالة الشرطية(IIf)لتضمين كود التعريف بشكل اختياري وفاصلته:

```
=Fields!Title.Value & " (#" & CStr(Fields!CopyNumber.Value) & ", " &
Fields!MediaName.Value & IIf(IsNothing(Fields!ItemBarcode.Value), "",", " & Fields!ItemBarcode.Value) & ")"
```

كما ذكرت سابقاً،حقل تاريخ استحقاق الإعادة due date لديه تعبير في خاصية اللون تحول لون الخط إلى أحمر عندما يكون البند متأخر الإعادةoverdue .

ReportOverdue.rdlc

يعمل هذا التقرير على إظهار قائمة بالبنود المتأخرة الإعادة في النظام.بما أن كل شيء سيكون متأخر،عملت على وضع حقل تاريخ الإعادة لأن يستخدم بشكل دائم لون الخط الأحمر.بدل العنوان،إن التقرير مطابق لتقرير البنود المخرجة(أو المعارة).

ReportMissing.rdlc

يعمل هذا التقرير على إظهار قائمة بجميع البنود المعلمة على أنه تم فقدانها.حتى ولو كان المخطط يتضمن حقل تاريخ استحقاق الإعادة،لن استخدمه في هذا التقرير.باقي التقرير مطابق بشكل أساسي لتقرير البنود المخرجة.

ReportPatronFines.rdlc

يعمل هذا التقرير على إظهار قائمة بجميع الزبائن الذين ما تزال عليهم غرامات مستحقة،وكمية الغرامة المستحقة.ويستخدم تخطيط الفئة ReportSchemaPatronFines.

الحقل الذي يعرض الغرامات لديه " C "في خاصية التنسيق Format.تنسيق الكود هذا يجبر قيمة عشرية decimal لعرض العملة currency باستخدام الإعدادات الإقليمية(أو الثقافية culture)على النظام المحلي.وخاصية التنسيق هذه تستخدم نفس الأكواد المستخدمة في الطريقة String.Format.

ReportStatistics.rdlc

يعرض التقرير الخامس إحصاءات السجلات من بعض الجداول في قاعدة بيانات المكتبة.وهذا هو التقرير الوحيد الذي يستخدم تخطيط الفئة ReportSchemaStatistics.

يعرض التقرير نفسه فقط نصين في كل سجل:اسم وقيمةname and a value. ويعتمد على الكود المستدعى لتنسيق خاصية الحقول تلك.

إضافة عارض تقرير.Adding a Report Viewer

حان الوقت لإضافة أداة MicrosoftReportViewer.بما أن أداة MicrosoftReportViewer واحدة يمكنها أن تعمل على إظهار أي نوع تقرير RDLC،سنعمل على إضافة فورم واحد لمعالجة جميع التقارير المبنية داخلياً(أو الجاهزة).

أضف فورم جديد وسميها ReportBuiltinViewer.vbإلى المشروع.وضع خاصية النص Textلها إلى "تقرير المكتبةLibrary Report" وخاصية WindowStateإلى " Maximized"،وأيضاً،حمل أيقونة المشروع (*Book.ico*)إلى الخاصية Icon.ستجد نسخة من هذا الملف في دليل المشروع.إذا أردت،تستطيع تحجيم الفورم لسبب ما فنقطة البداية من أجل التقرير(استخدمت 400 ,680)ولكن كل تقرير سيبدأ باستخدام الحجم الأكبر عندما يتم استخدامه.

أضف أداة MicrosoftReportViewerمسماة ReportContent إلى الفورم،وضع خاصية Dockإلى Fill. وضع الخاصية ShowBackButtonوالخاصية ShowDocumentMapButtonإلى خطأ False.

الكود الذي سنعمل على إضافته إلى هذه الفورم هو تشكيلة عن الكود الذي كتبناه سابقاً في هذا الفصل.وسيعمل الكود الذي يبدأ كل تقرير سيمرر لهذه الفورم اسم ملف التقرير RDLC،واسم تخطيط البيانات المستخدم،والبيانات الفعلية.بما أن هذه التقارير ستكون غير"نموذجية modeless "(أي تستطيع الحفاظ عليها مفتوحة بينما ما يزال بإمكانك استخدام الأجزاء الأخرى من برنامج المكتبة)،لانستطيع السماح للكود المستدعي الانتظار حتى يعمل المستخدم على إغلاق التقرير قبل أن نطرح بيانات التقرير.سنسمح للتقرير التخلص من البيانات نفسها.لعمل هذا،نحتاج إلى الحفاظ على مرجع إلى هذه البيانات.أضف العبارة التالية إلى فئة الفورم ReportBuiltinViewer.

```
Private StoreDataTable As Object
```

تذكر أن بإمكان التقرير استخدام تشكيلة من تنسيقات مورد البيانات،متضمنةً اتصالات قاعدة البيانات،المصفوفاتarrays ،والتجمعاتcollections. ستستخدم التقارير من 1 إلى 4 الحالة System.Data.DataTable وسيمرر التقرير الخامس تجمع قائمة شمولية generic List collection.الوقت المفضل للتخلص من البيانات عندما يتم إغلاق التقرير.أضف معالج الحدث التالي إلى الفورم،والذي يؤكد على البيانات التي تدعم عمليات التخلص قبل استدعاء الطريقة Dispose.

```
Private Sub ReportBuiltinViewer_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
        ' التخلص من البيانات
        If (TypeOf StoreDataTable Is IDisposable) Then
            CType(StoreDataTable, IDisposable).Dispose()
        End If
    End Sub
```

الكود الذي يعمل على فتح عرض هذه الفورم يمرر في قيم التقرير الأساسية من خلال طريقة عامة مسماة StartReport.أضف كودها حالاً:

```
Public Sub StartReport(ByVal whichReport As String, ByVal whichDataSchema As String, _
            ByVal whichData As Object)
        ' تشغيل واحدة من التقارير الجاهزة,المعامل النسبي الأول هو اسم ملف التقرير في التنسيق
        ' "Library.xxx.rdlc."
        ' بينما المعامل النسبي الثاني يوفر اسم التخطيط المستخدم,في تنسيق
        ' "Library_xxx."
        ' بينما المعامل النسبي الثالث هو البيانات الفعلية التي سيتم ربطها بالتقرير,
        ' والتي يجب أن تطابق التخطيط
        Dim customDataSource As New Microsoft.Reporting.WinForms.ReportDataSource
        ' ربط العارض,التقرير, والبيانات
        ReportContent.LocalReport.ReportEmbeddedResource = whichReport
        customDataSource.Name = whichDataSchema
        customDataSource.Value = whichData
        ReportContent.LocalReport.DataSources.Add(customDataSource)
        ' عرض التقرير
        StoreDataTable = whichData
        Me.Show()
    End Sub
```

يخبر هذا الكود العارض لاستخدام أي من التقارير كمصدر مضمن،ومن ثم إرفاق البيانات كمصدر بيانات مخصص. Local في أسماء الخاصيات يشير إلى تقرير العميل client محلي بدل تقرير "خادمserver " والذي يشتغل ضمن سكول سرفر.عندما كنا نشغل التقارير من قبل،كنا نرى أن نمط العرض الافتراضي كان نمط "ملئ كامل الشاشة بمحتوى الصفحة"،شخصياً،أفضل رؤية حدود تلك الصفحة المزيفة.لا تتضمن الأداة MicrosoftReportViewer خاصية تتيح لنا تغير العرض الافتراضي،ولكن نستطيع ضبط نمط العرض الأولي من خلال الطرق على هذه الأداة.عندما أضفنا عارض تقرير إلى الفورم،عمل الفيجوال أستوديو أيضاً على إضافة العبارة التالية لمعالج حدث تحميل الفورم Load.

```
ReportContent.RefreshReport()
```

أضف الكود التالي قبل هذه العبارة تماماً:

```
' إنتاج وعرض التقرير
        ReportContent.SetDisplayMode(Microsoft.Reporting.WinForms.DisplayMode.PrintLayout)
        ReportContent.ZoomMode = Microsoft.Reporting.WinForms.ZoomMode.Percent
        ReportContent.ZoomPercent = 100
```

إضافة تقارير جاهزة.Adding Built-in Reports

نسيت أنني منذ أمد عملت على إضافة الفورم *ReportSelect.vb* التي تقود التقارير،ولكنها موجودة في المشروع.في حال أنك نسيت كيف تبدو،سينعش الشكل التالي ذاكرتك. عملنا سابقاً على إضافة دعم من أجل خمس تقارير مبنية داخلياً(أو جاهزة) في كود هذه الفورم.نحتاج إلى إضافة كود غفلنا عنه سابقاً،إذا كنت تستخدم ملف تركيب تقرير XML لملئ قائمة التقرير،وعملت على توفير شرح من أجل كل تقرير في XML،تعرض كل مدخلة ذلك الشرح في النصف السفلي من نموذج اختيار التقرير*ReportSelect*. ولكن إذا كنت لاتستخدم ملف التركيب،وتعتمد فقط على الفورم لإضافة التقارير الخمسة الجاهزة بشكل افتراضي(التي عملتها)،فإن الفورم لن تعرض الشروحات المرافقة،لأننا نسينا إضافتها.أضف دالة إلى الفئة ReportSelect والتي تعود بوصف قصير لكل من التقارير الخمسة.

```
Private Function GetBuiltinReportDescription(ByVal whichReport As ReportItemEnum) As String
        ' العودة بالشرح المعرف سابقاً من أجل التقارير الجاهزة
        Select Case whichReport
            Case ReportItemEnum.BuiltInCheckedOut
                Return "بالاسم مرتبة,حالياً إخراجها تم التي البنود جميع تعرض"
            Case ReportItemEnum.BuiltInOverdue
                Return "بالاسم مرتبة ,الإعادة المستحقة البنود جميع تعرض"
            Case ReportItemEnum.BuiltInMissing
                Return "بالاسم مرتبة ,المفقودة البنود جميع تعرض"
            Case ReportItemEnum.BuiltInFinesOwed
```

```
                Return "الزبون اسم بواسطة مرتبة,زبون على مدفوعة والغير المستحقة الغرامات جميع تعرض"
            Case ReportItemEnum.BuiltInStatistics
                Return "المكتبة بيانات قاعدة من المسجلة الإحصاءات بعض تعرض"
            Case Else
                Return "التقرير لهذا شرح لايوجد"
        End Select
    End Function
```

سنعمل على استدعاء هذا الكود من مكانين.الأول في الطريقة LoadReportGroup.يعمل هذا الكود على تحميل في ملف تركيب تقرير XML.إذا كان ذلك الملف يتضمن واحد من التقارير الجاهزة،ولكن لايزود وصف معه،سنعمل على تزويد الوصف بأنفسنا.في منتصف ذلك الكود تقريباً ستجد هذه الأسطر.

```
'إذا,ماهو نوع المدخلة؟
                If (scanNode.Attributes("type").Value = "built-in") Then
```

وللأسفل بحوالي خمس أسطر ستجد العبارة التالية:

```
reportEntry.ItemType = CType(CInt(reportEntry.ReportPath), ReportItemEnum)
```

أضف الكود التالي بعد تلك العبارة تماماً:

```
If (reportEntry.Description = "") Then reportEntry.Description = GetBuiltinReportDescription(
reportEntry.ItemType)
```

الحاجة الثانية من أجل الشرح الجاهز يظهر في الطريقة RefreshReportList.تعمل هذه الطريقة استدعاء إلى LoadReportGroupلاستخراج تركيب XML. ولكن إذا بقيت قائمة التقرير بعد ذلك فارغة،تضيف RefreshReportListفي التقارير الخمس الافتراضية كل شرح مطلوب.قرب نهاية الطريقة،ضمن الحلقة For...Next،ستجد عبارة الإغلاق التالية:

```
'أضف مدخلة التقرير إلى القائمة
                AllReports.Items.Add(reportEntry)
```

أضف الكود التالي قبل تلك العبارة.

```
reportEntry.Description = GetBuiltinReportDescription(reportEntry.ItemType)
```

هذا كل شيء بخصوص وصف التقارير،والآن ارجع إلى كتابة التقارير الحقيقية. الكود الذي يعمل على بدء تشغيل كل من التقارير الخمسة موجود سابقاً في الفورم ReportSelectومعالج حدثها ActRun_Click.معظم الكود المضمن في العبارة Select Case والتي تعمل كلوحة تحكم من أجل التقرير المختار.إليك الجزء الذي يستدعي التقارير الخمس الجاهزة.

```
  Case ReportItemEnum.GroupLabel
'لاتقرير من أجل مدخلات مجموعة
                MsgBox("من فضلك اختر تقرير من القائمة", MsgBoxStyle.OkOnly Or MsgBoxStyle.Exclamation,
ProgramTitle)
            Case ReportItemEnum.BuiltInCheckedOut
'البنود تم استعارتها أو اخراجها
'للعمل:اكتب
                'BasicReportCheckedOut()
            Case ReportItemEnum.BuiltInOverdue
'البنود متأخرة
'للعمل:اكتب
                ' BasicReportOverdue()
            Case ReportItemEnum.BuiltInMissing
'البنود مفقودة
'للعمل: اكتب
                ' BasicReportMissing()
            Case ReportItemEnum.BuiltInFinesOwed
'الغرامات المستحقة على الزبون
'للعمل:اكتب
                ' BasicReportFines()
            Case ReportItemEnum.BuiltInStatistics
'إحصاءات قاعدة بيانات المكتبة
'للعمل:اكتب التقرير
                ' BasicReportStatistics()
            Case ReportItemEnum.ExeProgram
'بدء البرنامج
                Process.Start("""" & reportEntry.ReportPath & """ " & _
                    reportEntry.ReportArgs)
            Case ReportItemEnum.UrlProgram
'بدء عناوين الانترنت
                ' ----- Start a URL.
                Process.Start(reportEntry.ReportPath)
        End Select
```

من الواضح،أن هذا الكود لا يعمل الكثير.بدل كل من الأسطر "للعمل "،أزل الجزء"للعمل:اكتب" من العبارة.بحيث،يبدو مثلاً في السطر الذي يقول:

```
'للعمل:اكتب
                'BasicReportCheckedOut()
```

غير الكود إلى :

```
BasicReportCheckedOut()
```

أعمل ذلك لكل من الأسطر الباقية وذلك بإزالة التعليق عنها.

لعرض هذه الطرق الخمسة يتوجب علينا كتابة هذه الطرق.ستعمل هذه الطرق على استخلاص البيانات من التقرير،وترسل البيانات إلى عارض التقرير،على طول مع اسم الملف RDLC.وهي قصير جداً وبسيطة،لنعمل على إضافة هذه الطرق بالترتيب إلى الفئة ReportSelectالتي نحن بصددها الآن.لنبدأ أولاً بالطريقة BasicReportCheckedOut.

```
Private Sub BasicReportCheckedOut()
' تشغيل التقرير الأول الجاهز:تقرير البنود المخرجة
        Dim sqlText As String
        Dim reportData As Data.DataTable
        Dim reportForm As ReportBuiltinViewer
        On Error GoTo ErrorHandler
' استخلاص البيانات كمجموعة بيانات
        sqlText = "SELECT PA.LastName + ', ' + PA.FirstName AS PatronName, " & "PA.Barcode AS PatronBarcode, " & _
            "PC.DueDate, IC.CopyNumber, IC.Barcode AS ItemBarcode, " & _
            "NI.Title, CMT.FullName AS MediaName " & _
            "FROM Patron AS PA " & _
            "INNER JOIN PatronCopy AS PC ON PA.ID = PC.Patron " & _
            "INNER JOIN ItemCopy AS IC ON PC.ItemCopy = IC.ID " & _
            "INNER JOIN NamedItem AS NI ON IC.ItemID = NI.ID " & _
            "INNER JOIN CodeMediaType AS CMT ON NI.MediaType = CMT.ID " & _
            "WHERE PC.Returned = 0 " & _
            "AND PC.Missing = 0 " & _
            "AND IC.Missing = 0 " & _
            "ORDER BY NI.Title, IC.CopyNumber, PA.LastName, PA.FirstName"
        reportData = CreateDataTable(sqlText)
' اختبار عدم وجود البيانات
        If (reportData.Rows.Count = 0) Then
            reportData.Dispose()
            MsgBox("لايوجد بنود معارة", MsgBoxStyle.OkOnly Or MsgBoxStyle.Exclamation, ProgramTitle)
            Return
        End If
' إرسال البيانات إلى التقرير
        reportForm = New ReportBuiltinViewer
        reportForm.StartReport("Library.ReportCheckedOut.rdlc", "Library_ReportSchemaPatronItems", reportData)
        Return
ErrorHandler:
        GeneralError("ReportSelect.BasicReportCheckedOut", Err.GetException())
        Return
    End Sub
```

يعمل الكود على استخراج سجلات خاصة بالتقرير من قاعدة البيانات،ويتأكد من أنه تم تضمين سجل على الأقل.(يمكن أن نكون قد أضفنا عبارة سكول إلى قاعدة بيانات المكتبة إما كإجراء مخزن stored procedure أو عرضview، ونستدعيه بدل ذلك.من أجل أهداف هذا التدريب،كان من الأبسط تخزين العبارة مباشرة في الكود).ومن ثم يعمل على استدعاء عارض التقرير،ممرراً له اسم ملف RDLC،واسم التخطيط(في التنسيق ProjectName_ClassName)،وجدول البيانات.

التالي،أضف الطرق BasicReportOverdue و BasicReportMissing،لن أعمل على إظهار الكود هنا،بما أنه مشابه ما عدا اسم الملف RDLCوالشرط WHEREفي عبارة سكول،وهي متطابقة لـ BasicReportCheckedOut.

وأضف الطريقة BasicReportFinesأيضاً،والتي تعالج التقرير الجاهز الرابع .

إنه مشابه تماماً للطريقة BasicReportCheckedOut،ولكنها تستخدم عبارة سكول التي صممناها سابقاً من أجل استخراج غرامات زبون.ويستخدم أيضاً تخطيط مختلف واسم التقرير.

وأخيرا أضف الطريقة BasicReportStatistics إلى الفئة *ReportSelect.vb*، والتي تعالج التقرير الجاهز الخامس.وهو مختلف قليلاً عن الأربع الباقية لأنه يجمع البيانات من جداول ست مختلفة،واحد واحد،وفي كل حالة، يستخرج إحصاء لعدد السجلات في جدول قاعدة البيانات.ومن ثم يتم تخزين النتائج في تجميع شمولي(System.Collections.Generic.List)،حيث كل مدخلة قائمة هي حالة من ReportSchemaStatistics، الفئة المستخدمة من أجل تخطيط بيانات التقرير الخامس.(راجع كود هذه الطريقة وتمعن النظر فيها).

بما اننا نحتاج حقاً إلى الحصول على نفس المعلومات((*)COUNT) من أجل كل من الجداول الست المضمنة،عملت فقط على تنفيذ الكود كحلقة،وعملت على بناء عبارة سكول من أجل كل واحد كما مررته من خلال الحلقة.

تستطيع الآن تشغيل التطبيق وتستخدم التقارير الخمس الجاهزة.عليك الدخول كأمين مكتبة أو مدير،ومن ثم تمكّن من الوصول إلى لوحة "طباعة التقارير" على الفورم الرئيسية.

تقريباً عند نهاية هذا الفصل سنكون تقريباً قد انتهينا من التطبيق.الشيء الكبير المتبقي والذي علينا عمله معالجة بنود زبون المستحقة الماضية لرؤية فيما إذا كانت الغرامات مطلوبة، سنعمل على إضافة هذا الكود في الفصل التالي،وسنأخذ أيضاً نظرة عامة على الترخيص.

إجازة تطبيقك Licensing Your Application

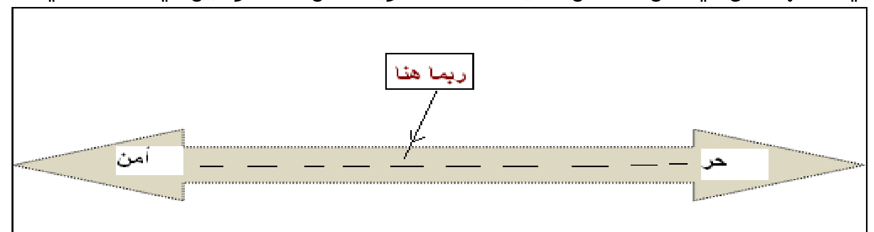
إجازة محتوى دوت نت المناسب يمكن أن يعني الاختلاف بين هيمنة السوق والإفلاس المالي،وما أحاول قوله فهم اتفاقية الترخيص التي تأتي مع الفيجوال أستوديو. سيبقى عليك استكشاف طرق الترخيص من أجل تطبيقاتك الخاصة قبل أن ترسلها إلى زبائنك.الترخيص و اتفاقيات الترخيص هي الوسائل الأساسية في حماية الملكية الفكرية التي عملت بجد لتطويرها.كيف يعمل الترخيص؟المفتاح موجودة في الكلمة نفسها:تأتي كلمة"ترخيص *license* "من " "-li"(رواية "أكاذيب")و " cense- "(من سنت cents كما في"بنس pennies ")مع بعضهما،هذان الجذران يعنيان"قص الأكاذيب حول قطع صغيرة من النقود".أدى التشويش في محاولة فهم ما تعني هذه العبارة إلى بقاء الفتيان السيئين في حيرة perplexedومشغولين occupiedلوقت طويل وكافي بحيث لا يسرقون تطبيقك.

إذا لم تعمل هذه الطريقة،فتوجد حلول برمجية ،بعضها سأعيد عرضه في هذا الفصل.يركز قسم من المناقشة على تصميم نظام الترخيص الذي سيظهر في مشروع المكتبة.يعمل إطار عمل الدوت نت على تضمين فئات من أجل ترخيص المكونات ولكنها تستخدم بشكل رئيسي من أجل مصممات الأدوات المستخدمة بواسطة مبرمجين آخرين ضمن بيئة التطوير المتكاملة للفيجوال أستوديوVisual Studio IDE .وليس من أجل تطبيقات المستخدم النهائية(أو إنهاء تطبيق المستخدم).لن نغطي ميزات الترخيص هذه في هذا الفصل.إذا كان لديك الفضول لمعرفة مثل هذه الميزات،ابدأ بقراءة "ما يخص مترجم الترخيص(*lc.exe*) License Compiler في المساعدة عبر الشبكة للفيجوال أستوديو.

خيارات ترخيص البرمجيات.Software Licensing Options

بالعودة إلى الأيام الأولى للبرمجيات،فلم يكن الترخيص بالقضية:فإذا كان بإمكانك الدخول إلى الكمبيوتر،فهذا لأنك كنت مرخص(مفوض لك).جميع تفاعل المستخدم مع النظام كان يتم من خلال المبرمجين واختصاصي التقنية.فإذا ما أراد مستخدم ما سرقة شيء ما،فسيكون هذا الشيء كأنه حمل 20 طن من الفولاذ،الأسلاك،وأنابيب التفريغ.أما الآن فكل شيء سهل.

اليوم القصة مختلفة.معظم المستخدمين غير مختصين بالتقنية،وبعضهم حتى غير أخلاقي،لذلك،لدينا الآن اتفاقيات الترخيص وفرق المحامين لمساندة كل هذا.ولكن لدينا أيضاً البرمجيات،البرمجيات التي بإمكانها فرض بعض القواعد بإحكام.من أجل جزء خاص من البرمجيات،ما يزال يوجد السؤال"ما مقدار كود التشديد على الترخيص الذي عليّ إضافته إلى التطبيق"؟كمية تحكم البرمجيات التي تضمنها ستقع في مكان ما ضمن الكمية المتسلسلة"حرية – أمن" كما هو مبين في الشكل التالي.



إذا ما ذهبت إلى النهاية الحرة للطيف(مناسب بالنسبة للمستخدمين والهكرز)،فستكون على حالة عالية من الثقة بمستخدمي تطبيقك،وأية حمايات مدعمة تعمل على إرسالها بسرعة dispatchedإلى مكاتبهم، لحفظ البرنامج في حالة امتثال(قابلية للتنفيذ بالنسبة لهم بحيث يستطيعون تجاوز هذه الحمايات).عند النهاية الآمنة من المسطرة المبينة في الشكل السابق("ضمانة بالنسبة للمبرمجين وشركات المحاماة العالية الأجر")تنفذ البرمجيات ممارسات وسياسات تضمن استخدام وحتى تنصيب التطبيق للمستخدمين المرخص لهم فقط،ليس هناك حاجة إلى الحمايات المدعمةarmed guards .

باقي هذا المقطع يناقش بعض الخيارات الممكنة التي تستطيع اختيارها ضمن المجال"حر – آمن".

اتفاقية الترخيص فقط .License Agreement Only

من الواضح أن الطريقة"اتفاقية الترخيص فقط" مفضلة بالنسبة للنمط الحر على الآمن،عندما تعمل على تزويد المستخدم بالبرنامج،يأتي مع اتفاقية ترخيص مصنوعة بحذر بحيث تنسق(تخطط) شروط الاستخدام لكل من المستخدم ومزودي البرنامج.بشكل عام تمنح المستخدم بعض الحقوق كالتنصيب،الاستخدام،ونشر البرنامج. عندما تكتب برنامج ليتم استخدامه من قبل منظمة معينة أو بواسطة مجموعة صغيرة من المستخدمين والذين سيكون لديك اتصال نظامي معهم،يمكن أن تكون اتفاقية الترخيص فقط هي ما تحتاجه حقاً.في الحقيقة،سأراهن أن معظم تطبيقات الفيجوال بيسك في هذا الاتجاه.لقد أعلنت ميكروسوفت منذ أعوام أن الغالبية العظمى من مبرمجي الفيجوال بيسك يستهدفون في استخدام تطبيقاتهم منظمات عمل معينة،مرتبطة بقاعدة بيانات خاصة معينة . تتطلب مثل هذه الأنظمة القليل جداً من طرق تشديد الترخيص،بما أن التطبيق غير نافع عند إخراجه خارج البناء الذي كان القصد من التصميم.حتى ولو حققت برمجياتك انتشار واسع،فإن خطة الترخيص يمكن أن تبقى هي الطريقة.العديد من التطبيقات المفتوحة المصدر،من ضمنها أنظمة التشغيل الرئيسية والتي تتناغم مع "بلينكس"(*http://www.fsf.org/licensing/licenses/gpl.html*)تستخدم ترخيص عام وشامل للأساس البرمجي الحر Free Software Foundation كسياسة رئيسية للنشر والترخيص.

مفتاح الترخيص العام المولد.Generated General License Key

إذا كنت تريد القليل من التحكم على النشر،التنصيب،واستخدام تطبيق ما،تستطيع فرض مفتاح الترخيص العام الناتج *generated general license key* – بشكل أساسي كلمة المرور passwordالتي تسمح بتنصيب التطبيق أو استخدامه.مثل هذه المفاتيح يتم إدخالها غالباً عند بداية تشغيل عملية التنصيب من قبل المستخدم حيث يتم الطلب من المستخدم مفتاح معين.بدون المفتاح لايمكن تنصيب التطبيق.

يحتاج بائعي البرمجيات إلى طريقة جيدة لتوليد مجموعة جيدة من مفاتيح التنصيب المميزة.يوجد زوج من الخيارات:

.إنتاج فقط رقم التسلسل التتابعيsequential serial number ،ومزج معرف المنتج ورقم الإصدار ضمنه.الشيء العظيم فيما يخص مثل هذه المفاتيح هي أنها سهلة التوليد.فلا يحتاج برنامج التنصيب عمل أي منطق تحقق معقد على المفتاح.ما يحتاج إليه فقط ضمان أن التنسيق العام صحيح.وإلى حدٍ ما،هذه الطريقة ليست أكثر أمناً من استخدام اتفاقية الترخيص فقط،بما أن أي واحد يعرف التنسيق العام بإمكانه صنع مفتاح خاص به.

.استخدام المفتاح المتمازج أو المخلوطhashed or scrambled key ،بالاعتماد على رقم تسلسلي ما أو صيغة يمكن أن يتم التحقق منها بواسطة برنامج التنصيب.يمكن أن تولد خوارزمية مصنوعة بشكل جيد مجال عريض من المفاتيح،ولكن تجعل من الصعب بالنسبة لآخرين لايعلمون الصيغة إنتاج مفاتيح مزورة خاصة بهم.على الرغم من أنني غير مطلع على privyعمليات ميكروسوفت الداخلية،تظهر هذه الطريقة لأن تكون الطريقة التي يتم استخدامها من أجل مفاتيح السيديات ذات الـ25 حرف،من ضمنها ذلك المزود مع الفيجوال أستوديو. على الرغم من أنه صعب تلفيق المفاتيح بدون أساس واقعي أو حقيقي،فالطبيعة العامة للمفاتيح تجعلها عرضة للمشاركة.من أجل بعض برمجياتها،تضم ميكروسوفت مفتاح السيدي مع عملية التسجيل المعتمدة على الهاتف أو عبر الشبكة لتحسين الأمن.

.تزويد مفاتيح متمازجة أو مشفرة hashed or encrypted key بالاعتماد على رقم التسلسل الذي يتم تزويده(بشكل سري) مع برنامج التنصيب أو وسيلة النشر.عندما يعمل المستخدم على إدخال المفتاح،يتم فك التشفير أو يتم تحضيره،ومن ثم يتم مقارنته مع رقم التسلسل.إذا وفقط إذا تطابقا سيتم إتمام تنصيب البرنامج بشكل مناسب.

مفتاح الترخيص المخصص المولد.Generated Custom License Key

مفتاح الترخيص المنتج المخصص مشابه للمفتاح المنتج العام،ولكنه يستخدم معلومات شخصية يتم تزويدها من قبل المستخدم كجزء من عملية التوليد،مثل هذه المفاتيح أكثر تفاعلية ،وتتطلب اتصال المستخدم النهائي بشكل خاص مع بائع البرنامج(أو التطبيق على موقع الويب) لإكمال عملية التنصيب.خلال عملية الشراء أو التنصيب،يقوم المستخدم بجعل معلومات معينة (مثل اسم المالك وتاريخ الشراء)متاحة لبائع البرنامج.ومن ثم يستخدم البائع تشفير عام لمفتاح خاص public-private key encryption (التشفير غير المتماثل asymmetric cryptography)إما لتشفير كامل أو لتوقيع رقمي للمعلومات ذات الصلة.يتم إعادة التوقيع المشفر فيما بعد للمستخدم النهائي من أجل التنصيب.تستخدم عملية التنصيب الحصة العامة من زوج المفتاح لضمان أن ذلك التوقيع صحيح.سنستخدم طريقة مفتاح الترخيص هذه في مشروع المكتبة.

مفتاح الترخيص مع ذاتية الهاردوير أو القفل.License Key with Hardware Identity or Lock

من أجل بائعي البرمجيات الكثيري الظنparanoid ،أو أولئك الذين لديهم أسباب منطقية لحفظ الزمام محكم على قاعدة التنصيب الخاصة بهم.توجد حلول تتضمن الوصول النظامي إلى الهاردوير(مكونات الكمبيوتر)أو الخدمات لتأكيد أن البرنامج المنصب سابقاً شرعي وصحيح.واحدة من الطرق الشائعة تستخدم الدونغلdongle (جهاز صغير لحماية البرامج من النسخdongle)بشكل نموذجي جهاز port-based device معتمد على منفذ يو إس بي USBبحيث أنه يجب على البرنامج الوصول إليه كل مرة يتم تشغيله.يوفر بائع البرنامج الدونغل مع البرنامج المرخص،ويمكن أن يشفره بحدود معتمدة على الاستخدام أو معتمدة على التاريخ.مع امتيازات الانترنت،لدى بائعي البرمجيات خيار التحقق من الوقت الحقيقي عبر الشبكة.كل مرة يبدأ البرنامج،من الممكن أن يتمكن من الوصول إلى موقع بائع معروف ليشارك في عملية التحقق من الاستخدام.مثل هذه الأنظمة تتيح مراقبة متواصلة للبرنامج من قبل البائعين والذي ومن المحتمل أن تكون لديهم أسبابهم العملية أو الحكومية لتحديد استخدام البرنامج.

من أجل واحد من مشاريع زبوني،يجب علي أن أتمكن من الوصول إلى موقع ويب مشارك آخر بشكل شهري وأُحمِل بيانات خاصة لاستخدامها مع برمجيات ذلك البائع.يطلب ذلك البائع أن أتمكن من الوصول دائماً إلى موقعه من جهاز خاص مع عنوان برتوكول انترنت معين IP address .وسيرفض تزويد البيانات إذا ما حاولت الاتصال من أي جهاز أخر .إذا ما احتجت حقاً لاستخدام عنوان انترنت جديد(إذا،على سبيل المثال ،ما غيرت موفر خدمة الانترنت)،يجب علي أن اسلم ورقة عمل للبائع تخبره بعنوان الانترنت الجديد.فيبدو مزعج ومثير للغضب،ولكن البيانات التي يزودها تكون مميزة وصحيحة،ويشعر أن لديه عمل ويحتاج إلى حماية ذلك الاستثمارinvestment .بما أن زبوني يحتاج البيانات،فليس لدي خيار إلا أن أزعن لإجراء التحقق الشهري.

الوصول المتحكم. Controlled Access

المستوى الأعلى من الأمن يتطلب عدم ثقة صريحة blatant distrust بالمستخدم ، من المحتمل أيضاً وجود سبب معقول لهذا. يمكن أن يجعل بائعي البرمجيات منتجاتهم متاحة لعدد محدد فقط من الزبائن،ومن ثم على أساس عقد فقطوكجزء من اتفاقية العقد،يوافق الزبون أن يكون لديه عضو في الجهاز الإداري مدرب في بيع البرمجيات على الموقع، يشغل ويعمل صيانة على التطبيق من أجل الزبون.على الأقل سيطلب البائع أن يكون واحد من موظفيه متاح بشكل مباشر للزبون متى تم استخدام التطبيق. في عالم التطبيقات الغير مصنوعة من أجل البيع(للطلب)،فيبدو من المفرط unconscionableأن يكون مثل هذا النظام متواجد .ولكن في حالات الخطر العالية،اهتمامات الأمن تطفو لهذا المستوى بحيث لا يتجرأ أي فريق أن يتلبس بالكامل مخاطر تنصيب واستخدام تطبيق هو ملك للغير.

على الرغم من أنني أحث على استخدام هذا النظام من أجل مشروع المكتبة،ولكن أظن أننا سنبقى على التخطيط الأصلي في توظيف مفتاح ترخيص مولد مخصص-custom generated license key.

اتفاقيات الترخيص.License Agreements

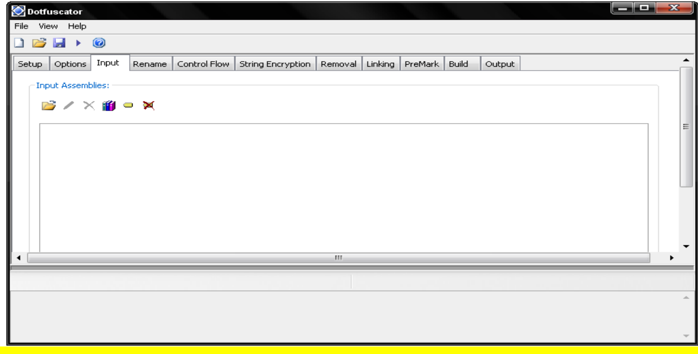
كثيراً ما ترى اتفاقية الترخيص تظهر قبل إتمام تثبيت برنامج ما.تخبر اتفاقية الترخيص المستخدم"اذهب للأمام،ونصب واستخدم البرنامج،ولكن عليك إتباع القوانين التالية."على الرغم من أنه يتم كتابتها غالباً بشكل قانوني،يمكن أن تظهر في اللغة الحقيقية،مثل الإنكليزية.وهي تختلف في مجال الحقوق الممنوحة،من "تستطيع استخدام هذا البرنامج،ولكن عندما تنتهي،يجب عليك تدمير جميع النسخ"إلى"استخدمه،وأنت حر في تمرير نسخة من البرنامج وكوده المصدري لأصدقائك وأقربائك."يأتي برنامج المكتبة الموفر مع هذا الكتاب مع اتفاقية الترخيص.

إن أي برنامج تعمل على صياغته ضمن إطار العمل لأن يتم استخدامه خارج شركتك الخاصة يجب أن يتضمن نوعاً ما من الاتفاقية بينك(شركتك)وبين مستخدم ا لبرنامج.يمكن لهذه الاتفاقية أن يتم تحديدها كجزء من العقد الذي يعمل على تأسيس مشروع تطوير البرنامج(وهذا نموذجي بالنسبة لمستشاري البرنامج)أو يمكن أن تعمل على تضمين الاتفاقية كمكون من البرنامج(شائع الاستخدام من أجل البرامج الغير موضوعة تحت الطلبoff-the-shelf programs).

مهما تكن الطريقة التي تختارها،من الهام أن تذكرها بالشكل المكتوب،لأنها يمكن أن تجنبك الندامة في المستقبل.توجد اتفاقية الترخيص عادةً لحماية حقوق بائع البرمجيات،ولكنها ستكون عديمة الفائدة إذا لم تمنح حقوق ذات أهمية للمستخدم-بعض الحقوق يمكن أن تكون كريمة جداً.

التشويش.Obfuscation

لقد لمحت على نحو قليل حول ميزات التشويش الموجودة في الفيجوال بيسك2008 في الفصل1 و5،ولكن حان الوقت لنلقي نظرة فعلية على هذه الميزات.تتضمن الفيجوال أستوديو إصدار ذو عرى(تبويبات)من Dotfuscatorتابع لشركة اسمها PreEmptive (ليست جزء من ميكروسوفت حتى الآن).لإمكانية الوصول إلى البرنامج،استخدم القائمة أدوات Tools>> Dotfuscator Community Editionفي الفيجوال أستوديو.تظهر الواجهة الرئيسية كما هو مبين في الشكل التالي.



ملاحظة: عند كتابة هذه الأسطر،لم يكن Dotfuscator Community Edition مضمن مع الفيجوال بيسك2008 بطبعته السريعةVisual Basic 2008 Express Edition .

حتى ولو كانت هذه النسخة الأساسية من المنتج،فإنك ترى أن لديها خيارات كثيرةgazillion. إذا كنت تريد الغوص إلى ميزات المحسنة من أجل مشروعك،فإن هذا غير ممكن.فسأغطي فقط الاستخدام الأساسي لها هنا.

دعنا نستذكر بشكل سريع لما تريد تشويش *obfuscate*كودك،أو حتى استخدام كلمة تشويش *obfuscate*في شركة مختلطة.إليك بعض الكود من مشروع المكتبة:

```
Public Function CenterText(ByVal origText As String, ByVal textWidth As Integer) As String
        ' ----- Center a piece of text in a field width.
        ' If the text is too wide, truncate it.
        Dim resultText As String
        resultText = Trim(origText)
        If (Len(resultText) >= textWidth) Then
            ' ----- Truncate as needed.
            Return Trim(Left(origText, textWidth))
        Else
            ' ----- Start with extra spaces.
            Return Space((textWidth - Len(origText)) \ 2) & resultText
        End If
    End Function
```

هذا الكود سهل الفهم تماماً،وخاصة مع التعليقات وأسماء الطرق والمتغيرات المعبرة.على الرغم من أن تشويش الدوت نت يعمل على مستوى لغة ميكروسوفت الوسيطة،دعنا نتظاهر أن ذلك المشوش يعمل مباشرة في كود الفيجوال بيسك.فتشويش هذا الكود يمكن أن يعمل على إنتاج نتيجة مشابهة للتالي:

```
Public Function A(ByVal AA As String, ByVal AAA As Integer) As String
        Dim AAAA As String
        AAAA = Trim(AA)
        If (Len(AAAA) >= AAA) Then
            Return Trim(Left(AA, AAA))
        Else
            Return Space((AAA - Len(AA)) \ 2) & AAAA
        End If
    End Function
```

في مثل هذا الروتين البسيط ما يزال بإمكاننا استكشاف(فهم) المنطق،ولكن مع جهد أكبر لما في النسخة الأصلية أعلى هذا الكود.بشكل طبيعي التشويش الحقيقي يذهب أبعد من هذا، مازجاً إمكانية القراءة للكود على مستوى اللغة الوسيطة،محيرة confoundingقارئي الكود والهكرز على حدٍ سواءalike. لتشويش مجمع مجمع assembly :

1.ابني مشروعك في الفيجوال أستوديو باستخدام قائمة بناء Build >>بناء[اسم المشروع] Build [Project Name].

2.شغل Dotfuscatorباستخدام قائمة أدوات Tools >> Dotfuscator Community Edition في الفيجوال أستوديو.

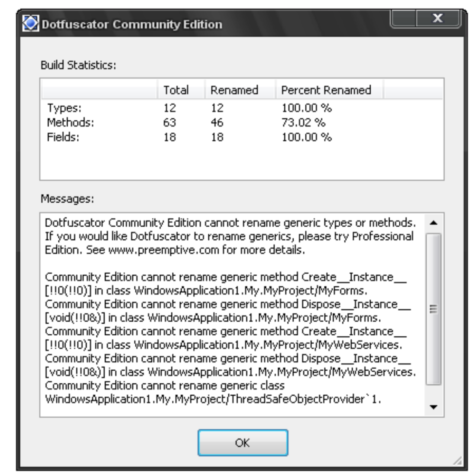
3.عند السؤال عن نوع المشروع اختر،إنشاء مشروع جديدCreate New Project، وانقر الزر موافقOK .

4.على تبويب "إدخال Input" لنافذة تطبيقDotfuscator، انقر الزر"استعراض وإضافة مجمع للقائمةBrowse and add assembly to list ".وهو الزر على شريط الأدوات في الزاوية اليسارية والذي يبدو ذو أيقونة مشابهة لمجلد ملفات مع سهم صغير في أعلاه-على اللوحة المبينة في الشكل السابق.

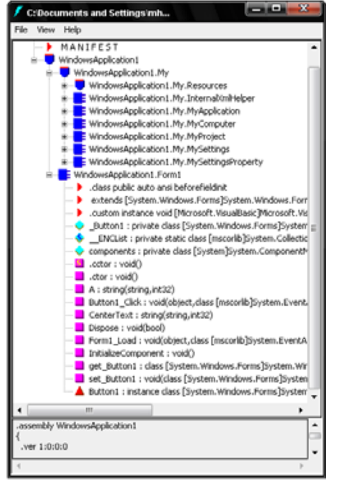
5.عند السؤال عند ملف المجمعassembly file، استعرض من أجل تطبيقك المترجم،وانقر الزر موافق OK. والمجمع الذي سيستخدم سيكون في الدليل الفرعي *bin\Release* ضمن دليل الكود المصدري لمشروعك.

6.اختر القائمة ملف File >>بناء Buildلإنتاج المجمع المبهمobfuscated assembly. سيتم سؤالك لحفظ ملف مشروع Dotfuscator(ملف XML)قبل أن تبدأ عملية البناء.احفظ هذا الملف إلى دليل جديدnew directory. عندما يحدث البناء ،فسيتم حفظ المجمع الناتج output assembly في الدليل الفرعي في نفس الدليل الذي يحتوي على ملف مشروع XML.

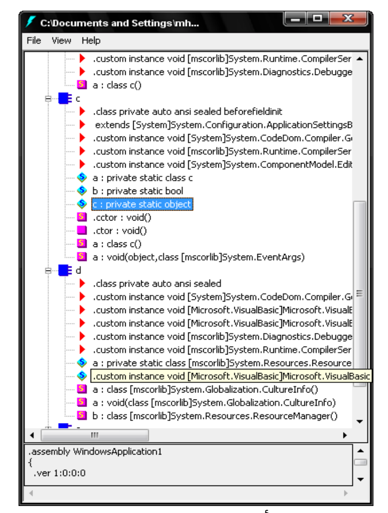
7.اكتمل البناء،ويظهر ملخص كما هو مبين في الشكل التالي.إن الملف المبهم جاهز للاستخدام.تعمل هذه العملية أيضاً على إنتاج ملف *Map.xml* f والذي يوثق تغيرات جميع الأسماء المعمولة للأنواع والأعضاء ضمن تطبيقك.وسيكون شيء سيء نشر هذا الملف مع المجمع.فهو من أجل أن تستخدمه أنت في تصحيحك فقط.



لبرهان أن التشويش قد حدث،استخدم أداة مفكك اللغة الوسيطة IL Disassemblerوالتي تأتي مع الفيجوال أستوديو لتفحص كل مجمع.(على نظامي،يمكن الوصول لهذا البرنامج بواسطة قائمة ابدأ Start >>جميع البرامج[All] Programs >>Microsoft Windows SDK v6.0A >>أدوات Tools >> Disassembler يبين الشكل التالي المتغيرات العامة المضمنة في ملف لمشروع المكتبة قبل الإبهام:



.النسخة المبهمة لهذه المتغيرات يظهر الشكل التالي



لن أعمل على إجراء تشويش على مشروع المكتبة من خلال مقطع التدريب هذا.وأنت حر في تجريبها فيما يخصك.

نظام ترخيص المكتبة.The Library Licensing System

الأدوات التي سنستخدمها لتصميم نظام ترخيص مشروع المكتبة يمكن أن يتم بناءه من Map.xml وهو ملفxml. وتم مناقشته في الفصول السابقة:

- يحتوى ملف الترخيص محتوى XML.الفصل13.
- يظهر الترخيص كملف منفصل في نفس المجمع *Library*.exe.يقرأ برنامج المكتبة المحتوى من ملف الترخيص.(الفصل15).
- سيتضمن الترخيص توقيع رقميdigital signature، معتمد على تشفير عام لمفتاح خاص public-private key encryption. الفصل11.

كلما تم تشغيل تطبيق المكتبة،يحاول قراءة ملف الترخيص.إذا كان الملف غير موجود،أو إذا كان يحتوي بيانات غير صحيحة أو توقيع غير صحيح،ينحدر البرنامج إلى الميزات الأدنى المتاحة،معطلاً تلك الميزات يتم اعتبارها مرخصة.

تصميم ملف الترخيص.Designing the License File

يحتوي ملف ترخيص مشروع المكتبة بعض الملكية الخاصة الأساسية ومعلومات الحقوق المتعلقة بالمستخدم الذي اشترى الحقوق ذات الصلة بالبرنامج.إليك محتوى XMLالذي جئت به:

```xml
<?xml version="1.0" encoding="utf-8"?>
  <License>
    <Product>Library Project</Product>
    <LicenseDate>1/1/2000</LicenseDate>
    <ExpireDate>12/31/2999</ExpireDate>
    <CoveredVersion>1.*</CoveredVersion>
    <Licensee>John Q. Public</Licensee>
    <SerialNumber>LIB-123456789</SerialNumber>
  </License>
```

يبدو أن هذا كافي.العملية التي تبني التوقيع الرقمي digital signature تخزن أيضاً توقيع مشفر ضمن محتوى XML.

إنتاج ملف الترخيص.Generating the License File

في مقطع المشروع لهذا الفصل،سنعمل على بناء تطبيق جديد يتواجد بشكل وحيد لتوليد ملفات الترخيص من أجل تطبيق المكتبة.سيكون له ثلاث مكونات رئيسية:
.إنتاج وإدارة المفاتيح العامة والخاصة المستخدمة في معالجة التوقيع.
2.الطلب من المستخدم تاريخ الترخيصlicense date ،تاريخ الانتهاء(انتهاء الصلاحيةexpiration date)،الإصدار المغطىcovered version ،اسم المرخص له licensee name،والرقم المتسلسل serial number من أجل ترخيص وحيد. وهي القيم التي تظهر في محتوى XMLلملف الترخيص.
3.إنتاج ملف الترخيص XMLوالتوقيع الرقمي digitally sign الذي يستخدم المفتاح الخاصprivate key .

تنصيب ملف الترخيص.Installing the License File

سيبين لك مقطع المشروع لهذا الفصل كيف يتم إنتاج ملف ترخيص شموليgeneric license. وهو ملف XMLسيتم نشره وتنصيبه مع تطبيق المكتبة باستخدام برنامج التنصيب الذي سنبنيه في الفصل 25. وسيتم تسمية الملف *LibraryLicense.lic*(بشكل افتراضي)وسيظهر دائماً في نفس الدليل لملف التطبيق *Library.exe* .
إذا كنت اعمل على تطوير تطبيق حقيقي من أجل بيعه للزبائن،ولدي موقع ويب،والذي يدعم خدمات ويب(التي سأتحدث عنها في الفصل23)،إليك واحد من التصاميم لتنصيب ملف الترخيص الذي يمكن أن استخدمه:
1.شغل برنامج التنصيب لتنصيب التطبيق على جهاز المستخدم.
2.أثناء التنصيب،يطلب برنامج التنصيب من المستخدم تفاصيل الترخيص التي تظهر بشكل نهائي في ملف ترخيص XML.
3.برنامج الإعداد يتصل بخدمة ويب على موقع الويب للبائع،ويمرر القيم الموفرة من قبل المستخدم لخدمة التسجيل .
4.تعمل خدمة الترخيص registration serviceعلى إعادة ملف XMLموقع بشكل رقمي يحتوي على محتوى الترخيص.
5.برنامج الإعداد setup program ينصب هذا الملف على طول مع التطبيق.
6.إذا لم يكتمل الترخيص بنجاح لأي سبب كان من خلال الإعداد،فإن التطبيق الرئيسي يحتوي كود ترخيص licensing code مشابه،ويمكن أن يتصل مع خدمة التسجيل registration service نفسها.

استخدام ملف الترخيص.Using the License File

كلما تم تشغيل تطبيق المكتبة،فإنه يقرأ ملف الترخيص XMLوينجز العديد من الاختبارات لضمان أن الترخيص متاح من أجل تنصيب التطبيق الحالي.إذا كان الترخيص غير صحيح لأي سبب كان،يمكن لمقاطع التطبيق الوصول إلى الميزات الإدارية المحسنة المضمنة في نظام المكتبة.

مشروع.Project

في كود مشروع هذا الفصل،سنتبع اثنين من أربع خطوات تم مناقشتها سابقاً في هذا الفصل في المقطع"نظام ترخيص المكتبة"،والمقطع :إنتاج واستخدام ملف الترخيص. والتصميم الذي عملنا على إنشائه سابقاً كافي وجيد بحيث يلبي حاجاتنا،على الرغم من أننا ما نزال بحاجة إلى تسجيله في التوثيق التقني للمشروع.لن نعمل على تنصيب ملف الترخيص رسمياً حتى نعمل على إنشاء برنامج التنصيب في الفصل25.

تحديث التوثيق التقني.Update Technical Documentation

بما أننا سنعمل على إضافة ملف خارجي جديد وسيتم معالجته بواسطة مشروع المكتبة،نحتاج إلى توثيق تركيبه في مجموعة الموارد التقنية للمشروع Technical Resource Kit.لنعمل على إضافة المقطع التالي إلى ذلك المستند.

ملف الترخيص.License File

يقرأ مشروع المكتبة ملف ترخيص يتم تعينه من قبل زبون مولد بواسطة تطبيق دعم منشئ ترخيص المكتبةLibrary License Generation support application. يعمل ذلك البرنامج على إنتاج ملف ترخيص XMLموقع رقمياً يتضمن معلومات صاحب الترخيص(أو المرخص له).إليك عينة عن محتوى ملف الترخيص:

```xml
<?xml version="1.0"?>
<License>
<Product>Library Project</Product>
<LicenseDate>1/1/2000</LicenseDate>
<ExpireDate>12/31/2999</ExpireDate>
<CoveredVersion>1.*</CoveredVersion>
<Licensee>John Q. Public</Licensee>
<SerialNumber>LIB-123456789</SerialNumber>
```

```
<Signature>
Digital signature appears here (not shown)
</Signature>
</License>
```

تشير العلامات<LicenseDate>و <ExpireDate> أول وأخر تاريخ للترخيص.وتشير العلامة <Licensee>إلى اسم صاحب الترخيص.وتتضمن العلامة <SerialNumber>رقم التسلسل المحدد من قبل البائع والمرافق مع الترخيص. وتحتوي العلامة<CoveredVersion> بيانات مشابهة لرقم نسخة المجمع المضمن في تطبيقات الدوت نت.ولديها أربع أجزاء محددة بنقطة:<major>.<minor>.<build>.<revision> . كل مكون يمكن أن يتضمن عدد من صفر0 إلى 9999،أو رمز النجمة(*)،والذي يشير إلى جميع القيم الصحيحة من ذلك الموضع.يحتوي المقطع <Signature>التوقيع الرقمي المولد.ويعتمد تنسيقه على أدوات تشفير XMLفي الدوت نت التي تولد ذلك المقطع.لضمان توقيع رقمي مناسب،استخدم دائماً تطبيق دعم منشئ ترخيص المكتبة لبناء ملفات الترخيص.

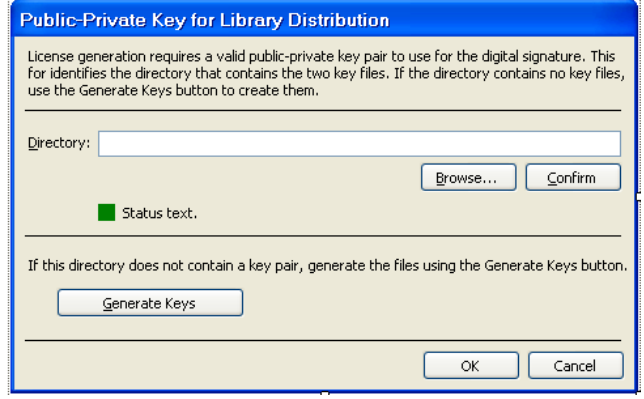
يعمل تطبيق الدعم على إنتاج زوج مفتاح عام وخاص من أجل الاستخدام في التوقيع الرقمي. الجزء العام من المفتاح(كــملف XML)يجب أن يتم إضافته كـــمورد مسمى LicensePublicKeyإلى تطبيق المكتبة.ويجب أن يبقى الجزء الخاص خاص.من أجل الاستقرار،يجب أن يتم استخدام نفس زوج المفتاح على طول عمر مشروع المكتبة المتوفر. سنعمل أيضاً على تخزين موضع ملف الترخيص كـإعداد للتطبيق في البرنامج الرئيسي.نحتاج إلى تسجيل ذلك الإعداد مع إعدادات التطبيق الأخرى التي عملنا على إضافتها إلى مقطع إعدادات المستخدم لمجموعة الموارد Resource Kit.

LicenseFileLocation

مسار ملف ترخيص المكتبة على محطة العمل هذه(أو الشبكة المحلية)إذا لم يتم توفيره،سيبحث البرنامج عن ملف مسمى *LibraryLicense.lic* في نفس مجلد التطبيق .

التطبيق المساعد لترخيص المكتبة.Library License Helper Application

سيكون إنتاج ملفات الترخيص والتواقيع الرقمية يدوياً باستخدام المفكرة سيكون... حسن،ولكننا لن نفكر بهذا.بالمقابل،سنعتمد على تطبيق خاص لإنتاج الملفات والتواقيع الخاصة بنا. عملت على إنتاج تلك الأداة الخاصة من أجلك،ستجدها في دليل تنصيب كود هذا الكتاب، في الدليل الجزئي (*LibraryLicensing*).يتضمن تطبيق الدعم هذا نموذجين رئيسيين.الفورم الأول(*KeyLocationForm*.*vb*) المبين في الشكل التالي يحدد أو يعمل على إنشاء ملفات مفتاح عام-خاص public-private key يتم استخدامها في عمليات التوقيع الرقمي digital signature .



يساعد معظم كود هذه الفورم على إيجاد والتحقق من المجلد الذي يحتوي على ملفات المفتاحين(واحد خاص،واحد عام).بعض الكود في معالج حدث ActGenerate_Clickيعمل على إنشاء الملفات الفعلية.

```
' ----- Generate and save the keys.
        Dim twoPartKey As RSA
        Dim currentStatus As KeyFolderStatus
        Dim publicFile As String
        Dim privateFile As String
' ----- Generate the keys.
        twoPartKey = New RSACryptoServiceProvider
        twoPartKey = RSA.Create()
' ----- Save the public key.
        My.Computer.FileSystem.WriteAllText(publicFile, _
            twoPartKey.ToXmlString(False), False)
        RefreshFolderStatus(KeyFolderStatus.MissingPrivateFile)
        ' ----- Save the private key.
        My.Computer.FileSystem.WriteAllText(privateFile, _
            twoPartKey.ToXmlString(True), False)
        RefreshFolderStatus(KeyFolderStatus.ValidFolder)
```

هذا بسيط جداً،فالفئة System.Security.Cryptography.RSA والفئة RSACryptoServiceProviderالمرتبطة بها تقومان بجميع العمل.كل ما عليك عمله استدعاء الطريقة RSA.Create،ومن ثم إنتاج مفاتيح XMLذات الصلة باستخدام الطريقة ToXmlString،ممرراً معامل نسبي (خطأFalse) من أجل المفتاح العامpublic key، وصواب True من أجل المفتاح الخاصprivate key. إذا كنت تريد الإطلاع على بعض عينات المفاتيح،افتح الدليل الجزئي (*LicenseFiles*).ستجد ملفين،واحد من أجل المفتاح العام والآخر من أجل المفتاح الخاص.سأعمل على طباعة واحد منهما هنا،ولكن سيبدو كحروف عشوائية.الفورم الأخرى هي الفورم *MainForm.vb*،والتي تعمل على إنتاج ملف الترخيص النهائي الفعلي للمستخدم،وتظهر في الشكل التالي.



كما مع الفورم الأولى،معظم الكود يضمن ببساطة أن ملفات المفتاح العام والخاص سليمة.وأن المستخدم قد أدخل بيانات صحيحة قبل الإنتاج. وأهم ما في هذه الفورم هو الزر"إنتاجGenerate" ومعالج حدثه ActGenerate_Click.أولاً،نحتاج إلى بعض محتوىXML ،والذي نبنيه بالطريقةBuildXmlLicenseContent. فهي تعمل على إنشاء المحتوى عنصر عنصر،باستخدام الطرق التي تحدثنا حولها في الفصل13،على سبيل المثال،إليك جزء من الكود الذي يعمل على إضافة الرقم المتسلسل.

```
' ----- Add the serial number.
        dataElement = result.CreateElement("SerialNumber")
        dataElement.InnerText = Trim(SerialNumber.Text)
        rootElement.AppendChild(dataElement)
```

ومن ثم يأتي التوقيع الرقمي،بواسطة الدالة SignXmlLicenseContent،معظمها يظهر هنا:

```
Private Function SignXmlLicenseContent(ByVal sourceXML As XmlDocument) As Boolean
        ' ----- Add a digital signature to an XML document.
        Dim privateKeyFile As String
        Dim privateKey As RSA
        Dim signature As SignedXml
        Dim referenceMethod As Reference
        ' ----- Load in the private key.
        privateKeyFile = My.Computer.FileSystem.CombinePath( _
            KeyLocation.Text, PrivateKeyFilename)
        If (My.Computer.FileSystem.FileExists(privateKeyFile) = False) Then
            MsgBox("Could not locate the private key file.", _
                MsgBoxStyle.OkOnly Or MsgBoxStyle.Exclamation, ProgramTitle)
            Return False
        End If
        privateKey = RSA.Create()
        privateKey.FromXmlString(My.Computer.FileSystem.ReadAllText(privateKeyFile))
        ' ----- Create the object that generates the signature.
        signature = New SignedXml(sourceXML)
        signature.SignedInfo.CanonicalizationMethod =SignedXml.XmlDsigCanonicalizationUrl
        signature.SigningKey = privateKey
        ' ----- The signature will appear as a <reference> element in the
        '       XML. The signature object can generate that automatically
        '       if we tell it to. It's more important later when trying to
        '       verify the signature. This transform says to ingore the
        '       signature itself when comparing the signature.
        referenceMethod = New Reference("")
        referenceMethod.AddTransform(New XmlDsigEnvelopedSignatureTransform(False))
        signature.AddReference(referenceMethod)
        ' ----- Add the signature to the XML content.
        signature.ComputeSignature()
        sourceXML.DocumentElement.AppendChild(signature.GetXml())
        ' ----- Finished.
        Return True
ErrorHandler:
        MsgBox("The license could not be digitally signed due to the following " & _
            "error:" & vbCrLf & vbCrLf & Err.Description, _
            MsgBoxStyle.OkOnly Or MsgBoxStyle.Exclamation, ProgramTitle)
        Return False
    End Function
```

يحدث التوقيع الرقمي بواسطة الفئة SignedXml(في فضاء الأسماء System.Security.Cryptography.Xml).تستخدم هذه الفئة طرق توقيع مختلفة،والتي اخترتها (XmlDsigCanonicalizationUrl)يتم استخدامها من أجل XMLنموذجي وتتجاهل التعليقات الموجودة.

يظهر هذا التوقيع كعلامات وقيم في مخرجات XML،المضافة من خلال العبارة AppendChildقرب نهاية الدالة.بما أننا لا نريد أن يتم أخذ التوقيع نفسه بعين الاعتبار عندما نبحث فيما بعد عن ملف XMLمن أجل المحتوى الصحيح، تضيف الفئة SignedXmlالتوقيع كعلامة <reference>. ويحدث هذا في الكود بإضافة كائن مرجعي Reference object والذي يتم برمجته من أجل تلك الأهداف. وتمت إضافته من خلال استدعاء الطريقة signature.AddReference.

حالما يكون لدينا توقيع في محتوىXML ،نكتبه كمخرجات إلى ملف محدد من قبل المستخدم بواسطة الطريقة القياسية XmlDocument.Save في معالج الحدث ActGenerate_Click.

```
licenseXML.Save(LicenseSaveLocation.FileName)
```

إليك عينة ملف ترخيصXM والذي يتضمن توقيع رقمي.وهذا هو الذي عملت على تضمينه في الدليل *LicenseFiles*.

```
<?xml version="1.0"?>
<License>
  <Product>Library Project</Product>
  <LicenseDate>1/1/2000</LicenseDate>
  <ExpireDate>12/31/2999</ExpireDate>
  <CoveredVersion>1.*</CoveredVersion>
  <Licensee>John Q. Public</Licensee>
  <SerialNumber>LIB-123456789</SerialNumber>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
      <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
      <Reference URI="">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
        <DigestValue>Dn6JYIBI/qQudmvSiMvuOvnVBGU=</DigestValue>
      </Reference>
    </SignedInfo>
<SignatureValue>NULghI4WbzDLroIcf2u9aoybfSjXPJRN50UMrCPYa5bup+c7RJnqTM+SzP4jmfJWPPs7pOvDC/fbdNYVMaoyXW0j
L3Lk8du3X4JXpW3xp9Nxq31y/Ld8E+RkoiPO6KRGDI+RRZ8MAQda8WS+L2fMyenRAjo+fR9KL3sQ/hOfQX8=</SignatureValue>
  </Signature>
</License>
```

يظهر التوقيع الرقمي كمحتوى مشفر(مشوش) ضمن العلامة<SignatureValue>.والآن إذا ما حاول أي شخص تعديل أي من قيم الترخيص،فإن الترخيص لن يطابق بعدها التوقيع،وكامل الترخيص سيصبح غير صحيح.

فبدل استخدام التوقيع الرقمي،يمكن أن نعمل فقط على تشفير ملف الترخيص بالكامل ضمن مفتاح خاص،وفيما بعد استخدم مفتاح عام لفك تشفيره واختبار محتواه.ولكنني أفضل التوقيع الرقمي،بما أنه يسمح لأي كان من فتح ملف الترخيص واختبار بارامترات(ثوابت) الترخيص نفسه بينما ما يزال مانعاً لأي تغيرات.

إضافة الترخيص إلى برنامج المكتبة.Adding the License to the Library Program

لنعود إلى تطبيق المكتبة.سيعمل البرنامج على ضبط سلوكه بالاعتماد على فيما إذا تم ترخيص أم لا.ولكن لصنع ذلك التصميم،هناك حاجة إلى ضمان أن محتوى ملف الترخيص صحيح ولم تتم محاولة تزويره.لعمل هذا،نحتاج إلى طريقة تعمل على فك (أو تفسر) التوقيع وتقارنه مع باقي الترخيص لضمان تطابقه.عملنا على بناء التوقيع باستخدام مفتاح خاص،وعلينا فكه باستخدام مفتاح عام.نستطيع تخزين مفتاح عام في ملفه الخاص خارج البرنامج،ولكن من المحتمل أن يتم فقدانه(كما يتم فقدان مفتاح حقيقي تماماً).بالمقابل،سنعمل على تخزين المفتاح العام كمورد للتطبيق،موجود خارجياً في مجلد موارد الكود المصدري.عملت على إضافة هذا المورد في التطبيق،وسميته LicensePublicKey.بواسطة هذا المفتاح المضمن في التطبيق،أي إنتاج للمفاتيح العامة والخاصة ستتطلب تعديل لهذا المصدر. نشير في الكود إلى محتوى XML بخصوص المفتاح العام باستخدام اسم مورده(أو اسم مصدره resource name):

```
My.Resources.LicensePublicKey()
```

بعض ميزات الأمن تستخدم الفئات الموجودة في فضاء الأسماء System.Security.Cryptography.Xml.وهذا الفضاء ليس واحد من الفضاءات التي يتم تضمينها بشكل افتراضي في تطبيقات الفيجوال بيسك الجديدة،لذلك سيكون علينا إضافته بأنفسنا.افتح نافذة خاصيات المشروع project properties واختر تبويب المراجع References. وتحت قائمة المراجع،انقر الزر "إضافةAdd"،ومن ثم اختر System.Security من تبويبNET. لنافذة إضافة مرجع Add Reference التي تظهر.

بما أن نافذة خاصيات المشروع مفتوحة،انقر فوق التبويب "إعدادات Settings"،وأضف إعداد نص جديد واستخدم النص LicenseFileLocation من أجل اسمه.سنستخدم هذا الإعداد لتخزين المسار لملف الترخيص.احفظ وأغلق نافذة الخصائص.

حاجات ترخيصنا العامة على طول التطبيق بسيط إلى حدما.نحتاج فقط معرفة الحالة الحالية لملف الترخيص،وأن يكون لدينا إمكانية الوصول للعديد من قيم الترخيص بحيث نستطيع عرض رسالة قصيرة حول الترخيص.ربما نحتاج إلى عمل هذا في أجزاء متنوعة من البرنامج،لذلك لنعمل على إضافة بعض الكود الشامل generic code إلى الوحدة البرمجية *General.vb*. افتح هذه الوحدة البرمجية الآن.

عند أعلى ملف الوحدة البرمجية،يتضمن الكود مرجع إلى فضاء الأسماء System.Security.Cryptography،بما أننا عملنا على تضمين كود يعمل على تشفير كلمة مرور المستخدم.ولكن هذا لايغطي مواد XMLالمصدرية أو القياسية.لذلك أضف عبارتي Importsجديدتين كما يلي.

```
Imports System.Xml
Imports System.Security.Cryptography.Xml
```

سنستخدم عداد للإشارة إلى حالة الترخيص.أضف هذا العداد الآن إلى الوحدة البرمجية General.

```
Public Enum LicenseStatus
       ValidLicense
       MissingLicenseFile
       CorruptLicenseFile
```

```
        InvalidSignature
        NotYetLicensed
        LicenseExpired
        VersionMismatch
    End Enum
```

لنعمل أيضاً على إضافة تركيب بسيط يعمل على وصل القيم المستخرجة من ملف الترخيص،أضف كوده إلى الوحدة البرمجية General:

```
Public Structure LicenseFileDetail
        Public Status As LicenseStatus
        Public Licensee As String
        Public LicenseDate As Date
        Public ExpireDate As Date
        Public CoveredVersion As String
        Public SerialNumber As String
    End Structure
```

بشكل افتراضي،يظهر ملف الترخيص في نفس دليل التطبيق،باستخدام الاسم .أضف ثابت شامل إلى الوحدة البرمجية General والذي يعرف هذا الاسم الافتراضي.

```
Public Const DefaultLicenseFile As String = "LibraryLicense.lic"
```

كل ما نحتاجه الآن بعض الكود لملئ التركيب LicenseFileDetail. أضف الدالة ExamineLicense التالية إلى الوحدة البرمجية General.

```
Public Function ExamineLicense() As LicenseFileDetail
        ' اختبار ملف ترخيص التطبيق,والإعلام عن ما في داخله.
        Dim result As New LicenseFileDetail
        Dim usePath As String
        Dim licenseContent As XmlDocument
        Dim publicKey As RSA
        Dim signedDocument As SignedXml
        Dim matchingNodes As XmlNodeList
        Dim versionParts() As String
        Dim counter As Integer
        Dim comparePart As String
        ' لنرى هل ملف الترخيص موجود
        result.Status = LicenseStatus.MissingLicenseFile
        usePath = My.Settings.LicenseFilelocation
        If (usePath = "") Then usePath = My.Computer.FileSystem.CombinePath( _
            My.Application.Info.DirectoryPath, DefaultLicenseFile)
        If (My.Computer.FileSystem.FileExists(usePath) = False) Then Return result
        ' محاولة قراءة الملف
        result.Status = LicenseStatus.CorruptLicenseFile
        Try
            licenseContent = New XmlDocument()
            licenseContent.Load(usePath)
        Catch ex As Exception
            ' خطأ
            Return result
        End Try
        ' تحضير مورد المفتاح العام للاستخدام
        publicKey = RSA.Create()
        publicKey.FromXmlString(My.Resources.LicensePublicKey)
        ' التأكيد على التوقيع الرقمي
        Try
            signedDocument = New SignedXml(licenseContent)
            matchingNodes = licenseContent.GetElementsByTagName("Signature")
            signedDocument.LoadXml(CType(matchingNodes(0), XmlElement))
        Catch ex As Exception
            ' ما يزال المستند غير صالح
            Return result
        End Try
        If (signedDocument.CheckSignature(publicKey) = False) Then
            result.Status = LicenseStatus.InvalidSignature
            Return result
        End If
        ' ملف الترخيص صحيح.استخرج أعضاءه
        Try
            ' احصل على اسم صاحب الترخيص
            matchingNodes = licenseContent.GetElementsByTagName("Licensee")
            result.Licensee = matchingNodes(0).InnerText
            ' الحصول على تاريخ الترخيص
            matchingNodes = licenseContent.GetElementsByTagName("LicenseDate")
            result.LicenseDate = CDate(matchingNodes(0).InnerText)
            ' الحصول على تاريخ الانتهاء
```

```
            matchingNodes = licenseContent.GetElementsByTagName("ExpireDate")
            result.ExpireDate = CDate(matchingNodes(0).InnerText)
            ' الحصول على رقم النسخة أو الاصدار
            matchingNodes = licenseContent.GetElementsByTagName("CoveredVersion")
            result.CoveredVersion = matchingNodes(0).InnerText
            ' الحصول على الرقم المتسلسل
            matchingNodes = licenseContent.GetElementsByTagName("SerialNumber")
            result.SerialNumber = matchingNodes(0).InnerText
        Catch ex As Exception
            ' هل ما يزال المستند غير صحيح
            Return result
        End Try
        ' الاختبار فيما إذا التواريخ خارج المجال
        If (result.LicenseDate > Today) Then
            result.Status = LicenseStatus.NotYetLicensed
            Return result
        End If
        If (result.ExpireDate < Today) Then
            result.Status = LicenseStatus.LicenseExpired
            Return result
        End If
        ' اختبار الاصدار
        versionParts = Split(result.CoveredVersion, ".")
        For counter = 0 To UBound(versionParts)
            If (IsNumeric(versionParts(counter)) = True) Then
                ' تنسيق الاصدار هو:
                ' major.minor.build.revision.
                Select Case counter
                    Case 0 : comparePart = CStr(My.Application.Info.Version.Major)
                    Case 1 : comparePart = CStr(My.Application.Info.Version.Minor)
                    Case 2 : comparePart = CStr(My.Application.Info.Version.Build)
                    Case 3 : comparePart = CStr(My.Application.Info.Version.Revision)
                    Case Else
                        ' رقم النسخة أو الاصدار غير صالح
                        Return result
                End Select
                If (Val(comparePart) <> Val(versionParts(counter))) Then
                    result.Status = LicenseStatus.VersionMismatch
                    Return result
                End If
            End If
        Next counter
        ' يبدو أن كل شيء على ما يرام
        result.Status = LicenseStatus.ValidLicense
        Return result
    End Function
```

هذا الكثير من الكود ولكن معظمه يعمل على تحميل واستخراج القيم من ملف XML للترخيص.وجزء اختبار التوقيع قصير نسبياً:

```
        publicKey = RSA.Create()
        publicKey.FromXmlString(My.Resources.LicensePublicKey)
            signedDocument = New SignedXml(licenseContent)
            matchingNodes = licenseContent.GetElementsByTagName("Signature")
            signedDocument.LoadXml(CType(matchingNodes(0), XmlElement))
If (signedDocument.CheckSignature(publicKey) = False) Then
            result.Status = LicenseStatus.InvalidSignature
            Return result
```

الكائن SignedXmlالذي استخدمناه أيضاً لإنتاج ملف الترخيص الأصلي يحتاج إلى معرفة بالضبط أي علامة XMLتمثل التوقيع الرقمي ضمنه.ستفكر أن وجود عنصر مسمى <Signature>سيكون كشف كبير،ولكن من المحتمل أن لايكون كذلك.على أية حال،حالما تعمل على إسناد تلك العقدة باستخدام الطريقةSignedXml.LoadXml، تستطيع استدعاء الطريقةCheckSignature، ممرراً لها المفتاح العام.إذا أعادت صواب،فإن الحال جيد،أعني أن الكود لايعلم أي شيء عنك،ولكن التوقيع جيد.

عرض الترخيص على الفورم"حول"Display the License on the About Form

عندما عملنا على إضافة الفورم"حول البرنامجAboutProgram "إلى المشروع،عملنا على تضمين أداة لصاقة مسماة LabelLicensed.وهي تعرض في الوقت الحالي وبشكل دائم "غير مرخص"،ولكن الآن لدينا أدوات لعض الترخيص المناسب،إذا كان متاح.افتح الكود المصدري لهذه الفورم،وأضف الكود التالي إلى بداية معالج حدث تحميل الفورم AboutProgram_Load.

```
        'تحضير الفورم
        Dim licenseDetails As LicenseFileDetail
        ' عرض صاحب الترخيص
        licenseDetails = ExamineLicense()
```

```
        If (licenseDetails.Status = LicenseStatus.ValidLicense) Then
  LabelLicensed.Text = "مرخص لـــ " & licenseDetails.Licensee & vbCrLf & "الرقم المتسلسل " & licenseDetails.SerialNumber
        End If
```

يبين الشكل التالي الفورم"حول البرنامج"في حال التشغيل مع التفاصيل المعروضة من ملف الترخيص.عندما أعمل على عرض الفورم "حول المشروع"مرةً أخرى،فإنها ستعرض"غير مرخص"،بما أن اختبار التوقيع فشل.كيف أعمل على اختبار الكود بشكل مبكر؟ عملت على نسخ ملف *LibraryLicense.lic* من الدليل الجزئي *LicenseFiles* ووضعت تلك النسخة في الدليل الجزئي *bin\Debug* للكود المصدري للمشروع.فيما بعد،ستكون قادر على وضع الملف في أي مكان تريد وتعمل استعراض من أجل الحصول عليه.

التأكيد على الترخيص.Enforcing the License

عند نقطة ما،سيكون للترخيص الغير صحيح تأثير سلبي على استخدام التطبيق.عندما يحدث ذلك،سيكون علينا منح المستخدم فرصة من أجل تصحيح المشكلة بإيجاد ملف الترخيص الصحيح.سنعمل ذلك من خلال الفورم الجديدة*LocateLicense.vb*، عملت على إضافة هذه الفورم سابقاً إلى المشروع،وهي تظهر في الشكل التالي.



تبدأ هذه الفورم في كودها المصدري باستدعاء دالتها العامةChangeLicense، والتي تعود بصواب إذا غير المستخدم الترخيص.معظم كود هذه الفورم يعمل على إدارة العرض، وإحضار تفاصيل أسباب صحة أو عدم صحة الترخيص باستخدام نتائج الدالة .فإذا كان الترخيص لأي سبب غير صحيح،فالنقر على الزر"إيجاد..." يتيح للمستخدم الاستعراض من أجل إصدار أفضل.

يعمل المتغير LocationModifiedالذي على مستوى الفورم على الرجوع إلى المستدعي كـبادئ من جديد من أجل إنعاش حالة الترخيص.من أجل تطبيق المكتبة بشكل خاص،لا أرى نقطة للتأكيد على الترخيص عند البدء.بما أنه ليس خطأ الزبون في أن تسرق المكتبة هذا العمل البرمجي الهام.بالمقابل ،عملت على تأجيل عملية التحقق حتى المدير أو أمين المكتبة يحاول الوصول إلى الميزات المحسنة للتطبيق.عندها،إذا فشل اختبار الترخيص،سيكون المستخدم قادر على استعراض الديسك(أو القرص)من أجل ملف ترخيص صحيح. أظن أن المكان الأفضل لإضافة اختبار الترخيص هو بعد أن يعمل المدير على كتابة كلمة المرور بنجاح مباشرةً،إذا اختبرنا قبل تلك النقطة،فستمنح الزبون العادي القدرة على استعراض القرص،والذي يجب أن لايحدث،لذلك افتح الكود المصدري من أجل الفورم *ChangeUser.vb*،وابحث عن معالج الحدث ActOK_Click واعمل على إيجاد التعليق "تسجيل دخول ناجح":

```
'تسجيل دخول ناجح
        LoggedInUserID = CInt(dbInfo!ID)
        LoggedInUserName = CStr(dbInfo!LoginID)
```

قبل مقطع الكود هذا أضف كود اختبار الترخيص التالي:

```
'لاتسمح بتسجيل الدخول إذا كان البرنامج غير مرخص
        Do While (ExamineLicense().Status <> _
        LicenseStatus.ValidLicense)
 سؤال المستخدم لما ينبغي عمله                      '
            If (MsgBox("لديك كان إذا ,الإداري الاستخدام أجل من" & " صحيحة بصورة مرخص غير التطبيق هذا " & _
            "الوقت في" & " الصحيح الترخيص ملف إيجاد تريد هل" & " الآن منه التحقق تستطيع,الصحيح الترخيص ملف
الحالي؟", MsgBoxStyle.YesNo Or _
            MsgBoxStyle.Question, ProgramTitle) <> MsgBoxResult.Yes) Then
                dbInfo.Close()
                dbInfo = Nothing
                Return
            End If
 الطلب من المستخدم تحديث الترخيص                     '
            Call LocateLicense.ChangeLicense()
            LocateLicense = Nothing
        Loop
```

هذا يمنح المستخدم عدد غير محدد من الفرص لإيجاد ملف الترخيص المناسب.حالما يتم التحقق من ملف الترخيص،يتقدم الكود إلى الأمام ويمكّن الوصول الإداري.

المعالجة اليومية لبند.Daily Item Processing

المجموعة الأخيرة الرئيسية من الكود التي سيتم إضافتها إلى مشروع المكتبة غير متعلقة بالترخيص،وبالرغم من ذلك فهي هامة:معالجة الغرامات من أجل البنود المتأخرة،سنعمل على إضافة طريقة مشتركة تقوم بالمعالجة،ومن ثم نستدعيها عند الحاجة على طول التطبيق.

أضف الطريقة DailyProcessByPatronCopyالجديدة إلى الوحدة البرمجية General:

```
Public Sub DailyProcessByPatronCopy(ByVal patronCopyID As Integer, ByVal untilDate As Date)
```

```
' يعمل هذا الروتين معظم العمل الأساسي في معالجة غرامات التأخير.كل روتينات المعالجة اليومية الآخرى تستدعي هذا الروتين في النهاية
        Dim sqlText As String
        Dim dbInfo As SqlClient.SqlDataReader
        Dim daysToFine As Integer
        Dim lastProcess As Date
        Dim fineSoFar As Decimal
        On Error GoTo ErrorHandler
' الحصول على جميع القيم الأساسية الضرورية لمعالجة هذه المدخلة
        sqlText = "SELECT PC.DueDate, PC.ProcessDate, PC.Fine, CMT.DailyFine " & "FROM PatronCopy AS PC
" & _
            "INNER JOIN ItemCopy AS IC ON PC.ItemCopy = IC.ID " & "INNER JOIN NamedItem AS NI ON
IC.ItemID = NI.ID " & _
            "INNER JOIN CodeMediaType AS CMT ON NI.MediaType = CMT.ID " & "WHERE PC.ID = " &
patronCopyID & _
            " AND PC.DueDate <= " & DBDate(Today) & " AND PC.Returned = 0 AND PC.Missing = 0 " & "AND
IC.Missing = 0"
        dbInfo = CreateReader(sqlText)
        If (dbInfo.Read = False) Then
' فقدان سجل نسخة الزبون.من المحتمل بسبب أن هذا البند لم يكن متأخر
' حتى الآن,أو تم فقدانه أو شيء ما صحيح مثل عدم يجب زيادة الغرامات
            dbInfo.Close()
            dbInfo = Nothing
            Return
        End If
' إذا كنا قد عالجنا هذا السجل من قبل من أجل هذا اليوم,فلا تفعل ذلك مرة أخرى
        If (IsDBNull(dbInfo!ProcessDate) = False) Then
            If (CDate(dbInfo!ProcessDate) >= untilDate) Then
                dbInfo.Close()
                dbInfo = Nothing
                Return
            End If
            lastProcess = CDate(dbInfo!ProcessDate)
        Else
            lastProcess = CDate(dbInfo!DueDate)
        End If
' الغرامات على هذا السجل.استكشف قيمتها
        daysToFine = CInt(DateDiff(DateInterval.Day, CDate(dbInfo!DueDate), untilDate) -
DateDiff(DateInterval.Day, CDate(dbInfo!DueDate), _
            lastProcess) - FineGraceDays)
        If (daysToFine < 0) Then daysToFine = 0
        fineSoFar = 0@
        If (IsDBNull(dbInfo!Fine) = False) Then fineSoFar = CDec(dbInfo!Fine)
        fineSoFar += CDec(dbInfo!DailyFine) * CDec(daysToFine)
        dbInfo.Close()
        dbInfo = Nothing
' تحديث السجل بالغرامات الأخيرة ومعالجة المعلومات
        sqlText = "UPDATE PatronCopy SET ProcessDate = " & DBDate(untilDate) & _
            ", Fine = " & Format(fineSoFar, "0.00") & " WHERE ID = " & patronCopyID
        ExecuteSQL(sqlText)
        Return
ErrorHandler:
        GeneralError("DailyProcessByPatronCopy", Err.GetException())
        Resume Next
    End Sub
```

يختبر هذا الكود السجلPatronCopy، السجل الذي يرمّز إعارة بند واحد من قبل زبون،لرؤية فيما إذا كان متأخر،وإذا كان كذلك،ما هي العقوبات penaltyالتي سيتم إضافتها إلى السجل.يتضمن كل سجل حقلProcessDate .لانريد أن نغرم الزبون مرتين في نفس اليوم من أجل بند متأخر واحد(ولن نفعل ذلك)،لذلك نستخدم ProcessDateللتأكيد على أي الأيام الغير محملة بالغراماتuncharged .

توجد عدة أمكنة على طول التطبيق حيث أننا نريد استدعاء روتين المعالجة هذا بدون مضايقة المستخدم.الظهور الأول في الفورمPatronRecord، الفورم التي تعمل على إظهار الغرامات المستحقة على زبون.قبل إظهار تلك القائمة تماماً،سنعمل على تنشيط كل بند تم إخراجه من قبل الزبون لضمان أننا نعرض معلومات الغرامات الأحدث.افتح الكود المصدري لتلك الفورم،اعمل على إيجاد معالج حدث تحميل الفورمPatronRecord_Load، وأضف الكود التالي،قبل استدعاء RefreshPatronFines(-1)الذي يظهر في منتصف هذا الروتين.

```
' التأكد من تحديث كل بند
        For counter = 0 To ItemsOut.Items.Count - 1
            newEntry = CType(ItemsOut.Items(counter), PatronDetailItem)
            DailyProcessByPatronCopy(newEntry.DetailID, Today)
        Next counter
```

حالة التأخير لبند يجب أن يتم تنشيطها أيضاً قبل إدخاله.افتح الكود المصدري للفورم الرئيسي MainFormواعمل على إيجاد معالج الحدثActDoCheckIn_Click. ستجد التعليق الذي يبدأ بـ" معالجة البنود المفقودة". قبل التعليق تماماً،ادخل الكود التالي:

```
' إحضار حالة بند وقد تم تحديثها
        DailyProcessByPatronCopy(patronCopyID, CheckInDate.Value)
```

تحتاج الإخراجات أيضاً تنشيط غرامات زبون،قبل السماح لزبون من معرفة فيما إذا كانت موجودة،في الحقيقة،أي غرامات مستحقة.مازلنا في الفورم الرئيسية ،انتقل إلى معالج الحدث ActCheckOutPatron_Click،وأضف التصريح التالي لأعلى هذا الروتين:

```
Dim dbTable As Data.DataTable
Dim dbRow As Data.DataRow
```

في هذا الروتين اعمل على إيجاد التعليق الذي يبدأ بـ" ' إظهر للزبون فيما إذا كانت عليه أي غرامات مستحقة".وكالعادة،أدخل الكود التالي قبل هذا التعليق:

```
' إحضار سجل الزبون وقد تم تحديثه
        sqlText = "SELECT ID FROM PatronCopy WHERE Returned = 0 " & _
        "AND Missing = 0 AND DueDate < " & DBDate(Today) & _
        " AND (ProcessDate IS NULL OR ProcessDate < " & _
        DBDate(Today) & ") AND Patron = " & patronID
        dbTable = CreateDataTable(sqlText)
        For Each dbRow In dbTable.Rows
            DailyProcessByPatronCopy(CInt(dbRow!ID), Today)
        Next dbRow
        dbTable.Dispose()
        dbTable = Nothing
```

بالإضافة إلى المعالجة التلقائية للغرامات،يسمح مشروع المكتبة أيضاً لمدير أو أمين المكتبة من عمل المعالجة اليومية لجميع بنود زبون عندما يريدون.وهذا يحدث من خلال لوحة "المعالجة اليومية"عل الفورم الرئيسية(شاهد الشكل التالي).



حالياً،لاتعمل هذه اللوحة أي شيء،لذلك دعنا نغير ذلك،المهمة الأولى هي تحديث حالة اللصاقة التي تظهر أعلى اللوحة.أضف طريقة جديدة مسماة RefreshProcessLocation إلى الفورم الرئيسية MainForm كما يلي:

```
Private Sub RefreshProcessLocation()
        ' لنرى هل تم تحديث المعالجة بالاعتماد على الموضع المختار
        Dim sqlText As String
        Dim dbInfo As SqlClient.SqlDataReader
        Dim lastDate As Date
        Dim newLocation As Boolean
        On Error GoTo ErrorHandler
        Me.Cursor = Cursors.WaitCursor
        ' البحث عن تاريخ كل موضع
        lastDate = #1/1/1900#
        newLocation = False
        sqlText = "SELECT * FROM CodeLocation"
        If (ProcessLocation.SelectedIndex <> -1) Then
            If (CInt(CType(ProcessLocation.SelectedItem, ListItemData)) <> -1) Then sqlText &= " WHERE ID = " & _
                CInt(CType(ProcessLocation.SelectedItem, ListItemData))
        End If
        dbInfo = CreateReader(sqlText)
        Do While dbInfo.Read
            If (IsDBNull(dbInfo!LastProcessing) = True) Then
                ' لم يتم عمل معالجة بعد على هذا الموضع
                newLocation = True
            Else
                If (CDate(dbInfo!LastProcessing) > lastDate) Then _
                    lastDate = CDate(dbInfo!LastProcessing)
            End If
```

```
        Loop
        dbInfo.Close()
        dbInfo = Nothing
        Me.Cursor = Cursors.Default
        ' تنشيط عرض إخراج التاريخ
        If (newLocation = True) Or (lastDate < Today) Then
            ' بحاجة إلى معالجة
            ProcessStatus.Text = "لم يتم تحديث المعالجة"
            ProcessStatus.ImageIndex = StatusImageBad
        Else
            ' تحديث
            ProcessStatus.Text = "تم تحديث المعالجة"
            ProcessStatus.ImageIndex = StatusImageGood
        End If
        Return
ErrorHandler:
        GeneralError("MainForm.RefreshProcessLocation", Err.GetException())
        Resume Next
    End Sub
```

يعمل هذا الكود اختبار على حقل قاعدة البيانات CodeLocation.LastProcessing, وأيضاً على جميع المواضع،أو من أجل الموضع المختار من قبل المستخدم،ويعمل على تحديث حالة العرض تبعاً لذلك.

يختار المستخدم موضع من أجل المعالجة بالقائمة المنسدلة"موضع المعالجة"ولكننا لم نعمل على إضافة أي كود يملأ هذه القائمة.اعمل على إيجاد الطريقة TaskProcess في الكود المصدري للفورم الرئيسية وأضف التصاريح التالية لأعلى هذه الطريقة:

```
'نمط المعالجة اليومية
        Dim sqlText As String
        Dim dbInfo As SqlClient.SqlDataReader
On Error GoTo ErrorHandler
```

ومن ثم أضف العبارات التالية إلى نهاية هذه الطريقة:

```
'تنشيط قائمة المواضع
        ProcessLocation.Items.Clear()
        ProcessLocation.Items.Add(New ListItemData("<جميع المواضع>", -1))
        ProcessLocation.SelectedIndex = 0
        sqlText = "SELECT ID, FullName FROM CodeLocation ORDER BY FullName"
        dbInfo = CreateReader(sqlText)
        Do While dbInfo.Read
            ProcessLocation.Items.Add(New ListItemData( CStr(dbInfo!FullName), CInt(dbInfo!ID)))
        Loop
        dbInfo.Close()
        dbInfo = Nothing
        RefreshProcessLocation()
        Return
ErrorHandler:
        GeneralError("MainForm.TaskProcess", Err.GetException())
        Resume Next
```

كل مرة يختار المستخدم موضع مختلف من القائمة ،نحتاج إلى تحديث حالة العرض.أضف الكود التالي إلى معالج الحدث ProcessLocation_SelectedIndexChanged.

```
Private Sub ProcessLocation_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) 
Handles ProcessLocation.SelectedIndexChanged
        ' تحديث الحالة بالاعتماد على الموضع الحالي
        RefreshProcessLocation()
    End Sub
```

تحدث المعالجة اليومية عندما ينقر المستخدم على زر"معالجة".أضف الكود التالي لمعالج الحدث ActDoProcess_Click.

```
Private Sub ActDoProcess_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles 
ActDoProcess.Click
        ' معالجة جميع الكتب التي تم إخراجها
        Dim sqlText As String
        Dim dbTable As Data.DataTable
        Dim dbRow As Data.DataRow
        Dim locationID As Integer
        On Error GoTo ErrorHandler
        Me.Cursor = Cursors.WaitCursor
        ' الحصول على قائمة جميع البنود التي ومن المحتمل تحتاج إلى معالجة
        sqlText = "SELECT PC.ID FROM PatronCopy AS PC INNER JOIN ItemCopy AS IC " & "ON PC.ItemCopy = 
IC.ID WHERE PC.Returned = 0 AND PC.Missing = 0 " & _
            "AND IC.Missing = 0 AND PC.DueDate < " & DBDate(Today) & " AND (PC.ProcessDate IS NULL OR 
PC.ProcessDate < " & DBDate(Today) & ")"
        If (ProcessLocation.SelectedIndex <> -1) Then
            locationID = CInt(CType(ProcessLocation.SelectedItem, ListItemData))
```

```
            If (locationID <> -1) Then sqlText &= " AND IC.Location = " & locationID
        Else
            locationID = -1
        End If
        dbTable = CreateDataTable(sqlText)
        For Each dbRow In dbTable.Rows
            DailyProcessByPatronCopy(CInt(dbRow!ID), Today)
        Next dbRow
        dbTable.Dispose()
        dbTable = Nothing
        Me.Cursor = Cursors.Default
        MsgBox("المعالجة إكتملت", MsgBoxStyle.OkOnly Or MsgBoxStyle.Information, ProgramTitle)
' تحديث تاريخ المعالجة
        sqlText = "UPDATE CodeLocation SET LastProcessing = " & DBDate(Today)
        If (locationID <> -1) Then sqlText &= " WHERE ID = " & locationID
        ExecuteSQL(sqlText)
' تحديث حالة العرض
        ProcessStatus.Text = "تم تحديث المعالجة"
        ProcessStatus.ImageIndex = StatusImageGood
        Return
ErrorHandler:
        GeneralError("MainForm.ActDoProcess_Click", Err.GetException())
        Resume Next
    End Sub
```

والآن لتجريب الكود،شغل البرنامج،اعمل على إيجاد ملف الترخيص الصحيح،وجرب الميزات الإدارية المختلفة.

والآن مع نهاية هذا الفصل فإن الكود الرئيسي تقريباً انتهى ومشروع المكتبة جاهز للعمل الآن،ولكن بقيت بعض الأمور البسيطة،مثل تعلم بعض الأشياء بخصوص آسبي دوت نت ASP.NET،وهو موضوع الفصل القادم.

تطوير الويب.Web Development

عندما اخترع السير تيم بيرنرز لي الشبكة العنكبوتية العالمية الواسعة في عام 1989، في الحقيقة لم تكن صفقة كبيرة.بما أن المصمم الرئيسي لبروتوكول نقل النصوص الفائقة HTTP(أو المدمجةHypertext Transport (or Transfer) Protocol) و HTML لم يظهر بعد.ولكن معظم التقنيات التي تدخل في تركيب ونقل صفحات الويب كانت موجودة منذ سنوات وحتى عقود.فلغة الترميز المعممة المعيارية *Standard Generalized Markup Language* (أو القياسيةSGML) (أساس لغة ترميز النصوص الفائقة *HyperText Markup Language*أو HTML)وأنظمة الربط المدمجة hyper-linking systems كانت موجودة منذ عام1960،ونقل البيانات المعتمدة على الانترنت Internet based transmission of data بين العملاء والمخدمين كانت شائعة بين شبكات الجامعة university campuses وبعض الأعمال.مانزال ونحن في بداية القرن ال21،والشبكة العالمية هي المركز لكثير من تقنيات الكمبيوتر والتي تجعل رأسي يدور.الفضل كله للسيد بيرنرز لي.

تروج ميكروسوفت للدوت نت على أنه نظام تطوير صفحات الويب والبرمجيات ذات الصلة.وفي الحقيقة إنه نظام عظيم.فعندما ندخل في الكود، سنجد حوالي 90% من ما تعمله لكتابة تطبيقات الويب في الفيجوال أستوديو مشابه لما تعمله عندما تكتب تطبيقات سطح المكتب.إنه سهل العمل،ونوع من المتعة،لذلك ومن المحتمل أنك تريد كتابة بعض البرامج باستخدامها.وهذا ما سنعمله في هذا الفصل.ولكن في البداية،دعنا نوجز مراجعة لما حدث في عالم مخدم-عميل client-server لاتصالات الشبكة العنكبوتية العالمية .

كيف يعمل الانترنت.How the Internet Works

قبل الدوت نت،كان تطوير تطبيقات "من أجل الشبكة"عبء ثقيل وممل.ولسبب مقنع:حيث أن الشبكة العنكبوتية العالمية لم يتم تصميمها كمنصة برمجة أو منصة معالجة المنطق.كانت بشكل أصلي كلها موجهة حول إرسال ملفات نصية بتنسيق معين من كمبيوتر واحد إلى آخر. لاتوجد لغات برمجة للتعلم،ولا منطق مخصص،فقط نصوص بسيطة،وربما صور رسومية ثنائية أو كلاهما.

مستعرضات الويب(أو الشبكة العنكبوتية) الأولية كانت في الحقيقة مجرد برامج نسخ ملفات عظيمة.وعندما تشغل مستعرض Mosaic (الذي كل ما كان في تلك الأيام)وتطلب صفحة ويب من كمبيوتر أخر،إليك ما كان يحدث:

1.يحدد مستعرض الويب عنوان برتوكول الانترنت IP address للنظام البعيد.

2.يتصل مستعرض الويب بالنظام البعيد بواسطةtransmission control protocol/Internet protocol (أو برتوكول التحكم في الإرسال/برتوكول الانترنت) المنفذ ذو الرقم 80 .

3.يقبل النظام البعيد الاتصال.

4.يقول مستعرض الويب"مرحباً،إني أبحث عن ملف مسمى.*index.html* .هل تستطيع إرساله لي؟"

5.يقول النظام البعيد"إنه لدي".ومباشرةً يرسله له.

6.يعمل النظام البعيد على إغلاق الاتصال.

معظم هذه المعالجة مخفية ،ولكن تستطيع رؤية حدوثها.إذا كنت مهتم،افتح موجه أوامر ويندوز،واكتب الأمر التالي:

```
telnet www.google.com 80
```

هذا يشغل برنامج تيلنت network protocol(برتوكول انترنت)،أو برنامج المضاهاة الطرفي terminal emulation program الذي يتيح لك الاتصال إلى الأنظمة البعيدة من خلال واجهة نصيةtext interface. (يتم تنصيب Telnetعلى نظام ويندوز XPبشكل افتراضي،ولكنه اختياري في نظام فيستا،تستطيع إضافته إلى فيستا من خلال لوحة التحكم وميزات إضافة مكونات ويندوز.)عادة،يتصل تيلنت إلى المنفذ23،ولكن تستطيع تعين أي منفذ portتريد،كما فعلنا هنا مع المنفذ الافتراضي للشبكة العنكبوتية العالمية وهو 80.

يمكن أن تبدو شاشتك فارغة،أو ربما تبقى في مكانها،وتبدو كالجامدة،إذا كنت محظوظ،سترى رسالة "الاتصال"،ويمكن أن لا تراها.وكل شيء على مايرام.يعمل نظامك على الاتصال بمخدم الويب التابع لغوغل.اكتب الأمر التالي:

```
GET / HTTP/1.0
```

اتبع هذا السطر بضغطتين خفيفتين على مفتاح إنتر.يطلب هذا الأمر من النظام البعيد إرسال صفحة الويب الافتراضية عند أعلى التسلسل الهرمي لمخدم الويب ذاك.ولأنك سألت فسيكون:

```
HTTP/1.0 200 OK
Cache-Control: private
Content-Type: text/html; charset=ISO-8859-1
Set-Cookie: PREF=ID=1c1dd342e463f3f1:TM=1199325226
:LM=1199325226:S=Pl-4f1fg4yh8Mvw7;
expires=Sat, 02-Jan-2010 01:53:46 GMT;
path=/; domain=.google.com
Server: gws()
Date: Tue, 01 Jan 2008 01:30:00 GMT
Connection: Close()
<html><head>
...rest of HTML web page content here...
</body></html>
Connection to host lost.
```

بالطبع لاترى عادة كل هذا.يعمل مستعرض الويب على تحمل هذا الحوار عنك،وينسق بشكل جميل الاستجابة كصفحة ويب.هذا عملياً كل ما هنالك بخصوص الشبكة العنكبوتية العالمية.حيث أنك جربت فقط الميزات الرئيسية المضمنة:نقل البيانات الأساسية من خلال منفذTCP/IP . لذلك،أين تدخل البرمجة؟.

برمجة الانترنت.Programming the Internet

كانت الصفحات الستاتيكية(الساكنة)جيدة لفترة من الزمن.ومن ثم أصبح الانترنت ممل.أخيراً،جاء أحدهم بفكرة ساطعة:"لدينا برنامج يعمل على مخدم الويب ويستجيب للزبائن،ويغذيهم بالصفحات المطلوبة.ما الذي يحصل إذا ما استطعنا تحسين ذلك البرنامج ،بحيث من أجل صفحة معينة،سيستدعي برنامج أو صيغة ستعمل على توليد محتوى HTMLأثناء التشغيل أو المعالجة،ونعمل على إرجاع ذلك المحتوى إلى العميل؟"لذلك،عملوا على تغير معالجة المخدم.والآن،عندما يطلب العميل صفحة الويب التي تنتهي بالامتداد*.cgi* عملية مخدم الويب تشغل

صيغة منفصلة تعمل على توليد المحتوى.و النظام أيضا مزود بوسائل من أجل محتوى موفر من قبل العميل ليشق طريقه إلى الصيغة، جاعلاً من الممكن تخصيص وتطبيع الميزات.
من هناك كانت الخطوة القصيرة للحل الشامل.على منصة ميكروسوفت،الوظائف الإضافية add-ins المدعومة بخادم معلومات الانترنت *Internet Information Server*التي يمكن استدعاءها بالاعتماد على امتداد الملف للملف المطلوب.وهذا يقود إلى صفحات الخادم النشيطة *Active Server Pages* (ASP)، حل يتيح للمطورين تضمين صيغة من قبل الخادم(غالباً باستخدام صيغة الفيجوال بيسكVBScript، تشكيلة ما لفيجوال بيسك) في محتوى HTMLتماماً،وعليه تعديل المحتوى قبل أن يتم إرساله إلى العميل.
قال شخص آخر"إذا كان بإمكاننا كتابة صيغ scriptsعلى جهة الخادم،ألا نستطيع تضمين"صيغة العميل(الصيغة المكتوبة من قبل العميل)"تماماً في محتوى HTMLبحيث يمكن لمستعرض ويب حاذق معالجتها؟"قبل زمن ،كان كل من مطوري جهة العميل client-side وجهة الخادم server-side في حالة حرب معلنة في الشوارع،ولكن المعركة لم تستمر لفترة طويلة لأن معظم المبرمجين كانوا قد استنفذوا طاقاتهم exhausted.السبب؟البرمجة في صيغةscript programming !فإما تضمين الصيغة في HTML(جهة العميلclient side)أو توليد HTMLمن صيغة(جهة الخادمserver side)البرمجة بالصيغة مرهقة،بطيئة،"سيئة"،وغالباً من المستحيل التصحيح بشكل تفاعلي.
بعض مبرمجي الصيغة لم يستخدموا مترجم اللغة لسنوات،وكانوا على حافة السقوط the verge of lapsing في غيبوبة إغراء الصيغة المميت fatal script-induced comas.تستطيع ترجمة بعض المنطق المتصل بجهة الخادم إلى "مكتبة الربط الديناميكية.dynamic linked library "أو DLLوتستخدمه لمعالجة صفحات الويب،ولكنه بعيد عن السهولة،وهذه DLLsستبقى غالباً مربوطة إلى محتوى HTMLبواسطة الصيغ القصيرة.من ثم جاء الدوت نت ودعمه لتطوير تطبيق من جهة الخادم المترجم.تنفس مبرمجي الصيغة الصعداء ،فبإمكانهم الآن استخدام الطاقة الكاملة للغات الدوت نت والفيجوال أستوديو لبناء محتوى HTML.وهذا النظام الجديدASP.NET ,،تم تصميمه بحيث تستطيع صنع تطبيق ويب بالكامل دون حتى النظر إلى سمة HTML واحدة.هدف التصميم:جعل تطوير الويب مشابه تقريباً لتطوير تطبيق سطح المكتب.ونجحت ميكروسوفت على نحو عظيم.فهي لم تحل مشكلة صيغة العميل(وربما قريباً)،ولكن بعض ميزات جهة العميل الجديدة المضمنة في ASP.NET تخفض على نحو عظيم الحاجة إلى صيغ جهة العميل الخاصة.
الصفحات التي تبنيها في ASP.NET تدعى نماذج الويبWeb Forms ،ولأنها مربوطة بإحكام مع بعضها البعض،فاستخدم في بعض الأحيان ASP.NET ونماذج الويب بشكل تبادلي. ولكنهما ليسا نفس الشيء بالضبط:فـ ASP.NET يتضمن نماذج الويبWeb Forms .

ميزات ASP.NET . ASP.NET Features

يتضمن ASP.NETالعديد من تقنيات تطوير الويب المتقدمة الجديدة.إليك فقط القليل من التقنيات الأكثر شيوعاً:

الكود المترجم.*Compiled code*

كل الكود الذي تكتبه من أجل تطبيقات آسبي دوت نت ASP.NET مترجم بالكامل إلى مجمعات مكتبات الربط الديناميكية للدوت نت .NET DLL assemblies.عندما يطلب العميل ملف بالامتداد*aspx*.،يعمل خادم معلومات الانترنت Internet Information Server على إيجاد هذا الملف(والذي يحتوي محتوى HTMLأو محتوى HTML/ASP.NET.)و DLL المترجم المرفق،ويستخدمهما مع بعض لمعالجة محتوى صفحة.تستطيع عمل ترجمة سابقة لـ DLLقبل نشره.أو تستطيع ترك ASP.NET أن يعمل على ترجمته أثناء معالجة الملف*aspx*. للمرة الأولى التي يتم استدعاءه فيها(يسبب هذا أيضاً القليل جداً من إزعاج الأداء).

دعم الدوت نت.*.NET support*

يمكن لتطبيقات آسبي دوت نت ASP.NET الوصول إلى مكتبات فئات إطار عمل الدوت نت.NET Framework Class Libraries بالكامل(FCLs)، ما عدا تلك التي تستهدف تطوير تطبيقات سطح المكتب.فأياً من الميزات والفئات التي لديك في تطبيقات الدوت نت لسطح المكتب،موجودة في تطبيقات الويب أيضاً.

الاعتماد على الكائن.*Object-based*

وسمات HTML،مثل وسمة <textarea>،هي في الحقيقة مجرد نصوص(أو سلاسل حرفية) ضمن ملف HTMLنصي أكبر.يعامل آسبي دوت نت عناصر صفحة ويب ككائنات حقيقية،مجهزة complete with بالخاصيات propertiesوالأحداثevents .وبعض هذه الكائنات ينفذ أدوات جهة عميل معقدة،يتم تجهيزها بالمئات من أسطر صيغة جهة العميل client-side script التي تحصل عليها بالمجان.

البساطة في التوزيع(أو النشر).*Deployment simplicity*

إدارة صيغ جهة الخادم و DLLsالخاصة قبل الدوت نت لم تكن بهذه السهولة.إنشاء أنواع من التغيرات يتطلب إيقاف كامل لخادم معلومات الانترنت، أو على الأقل الحصة التي تتحكم بكون التطبيق قد تم تغيره.يتيح لك آسبي دوت نت بعمل تغيرات على نظام الإنتاج دون التأثير على عمل المستخدمين.إذا بدلت DLL المترجم سيعمل ASP.NET على بدء استخدامه مباشرةً،ولكن سيبقى محافظ على النسخة القديمة حتى ينفصل detachedجميع العملاء الموجودين منها.

استقلالية المستعرض.*Browser independence*

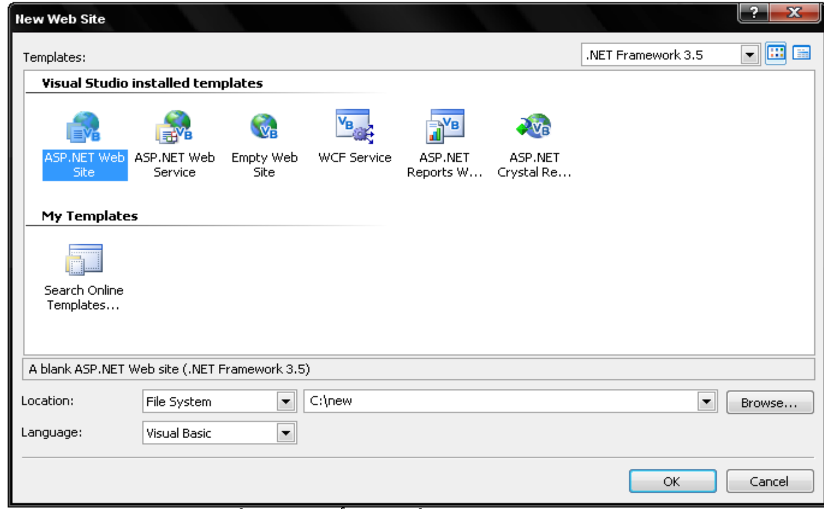
كائنات صفحة الويب التي تستخدمها في آسبي دوت نت تتولى مسؤولية إنتاج HTMLومحتوى صيغة جهة العميل.العديد منها تأخذ في حسبانها نوع مستعرض العميل وإصداره،محسنةً أو مخفضةً الميزات بشكل آلي عند الحاجة.وكمطور آسبي دوت نت،ليس عليك حتى معرفة المستعرض الذي سيتم استخدامه.

قابلية التوسيع.*Extensibility*

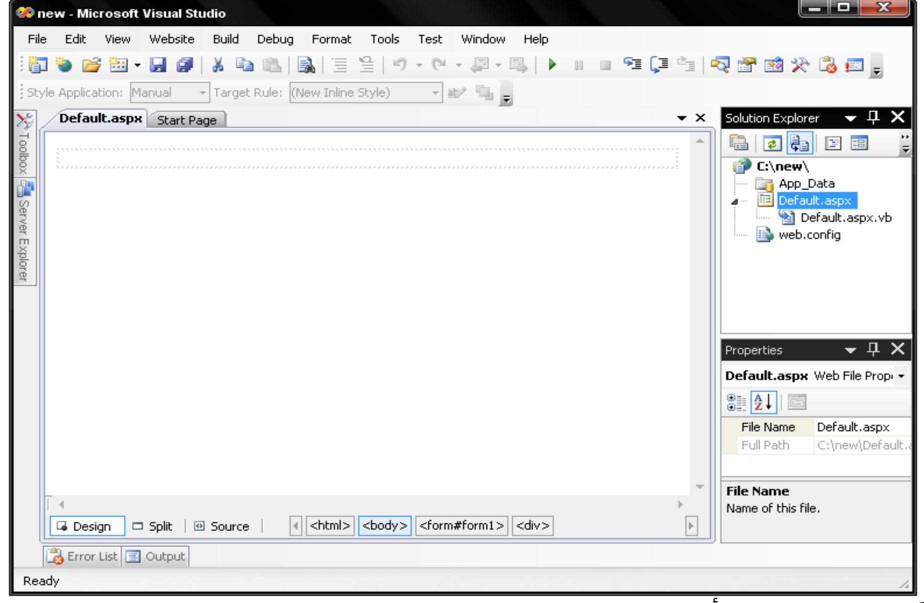
إذا كنت تريد تحسين عنصر صفحة ويب،تستطيع الاشتقاق من فئته وإضافة الميزات المحسنة الجديدة،تماماً كما تعمل مع فئات الدوت نت الأخرى.بالطبع توجد ميزات عظيمة أكثر من هذه القليلة التي جدولتها هنا.ولكن من المحتمل أنك تريد رؤية ASP.NET في حالة العمل،دعنا نبدأ.

تجريب آسبي دوت نت.Trying Out ASP.NET

لنعمل على بناء تطبيق ASP.NET بسيط .ونتفحصه وأجزاءه لاكتشاف كيف يبدو.
ابدأ الفيجوال أستوديو واختر قائمة ملف File>> موقع ويب جديدNew Web Site .يظهر نموذج الموقع الجديد(شاهد الشكل التالي).على خلاف تطبيقات سطح المكتب،يجب عليك أن تخبر الفيجوال أستوديو مباشرةً أين ستعمل على تخزين الملفات.سنختار موقع على نظام الملفات المحلي،ولكن يتيح لك هذا النموذج أيضاً العمل على موقع ويب بعيد بواسطة برتوكول نقل الملفات(FTP)file transfer protocol أو HTTP.أختر قالب موقع ويب لأسبي دوت نتASP.NET Web Site ،أدخل مسار الدليل حيث تريد أن تخزن الملفات،وانقر الزر موافقOK .



يبين الشكل التالي بيئة التطوير بعد النقر على زر موافق في الشكل الأعلى وأشرطة الأدوات المعروضة هي كل ما أفضل ظهوره. لوحة مستكشف الحلول تبين ثلاث ملفات ومجلد تم تضمينهم في المشروع.إذا ذهبت إلى دليل المشروع،الموقع الذي عملت على تحديده في الشكل العلوي،سترى نفس هذه الملفات الثلاثة. الملف *web.config* هو ملف XMLيحتوي على الإعدادات الخاصة بالتطبيق،وهو على صلة بالملف *app.config*المستخدم في تطبيقات سطح المكتب.إن الملف *Default.aspx* هو صفحة الويب نفسها،والتي ستحتوي على مزيج من HTMLوسوم آسبي دوت نت ASP.NET tags الخاصة والتوجيهات(التعليمات directives).والملف *Default.aspx.vb* ذو الصلة يحتوي كود الفيجوال بيسك"الكود المخفي" خلف الكود المصدري الذي سنعمل على ترجمته إلى DLL.



تعمل الفيجوال أستوديو أيضاً على إنشاء مجلد أخر *\Users\username\Documents\VisualStudio 2008\Projects\WebSite1*.:يحتوي هذا المجلد على ملفات الحل التي يتم إنشاءها عادة مع مشروع الفيجوال بيسك. حيث يتم وضعها خارج المسار بحيث لايتم تضمينها مع موقع الويب الموزع.المنطقة الفارغة التي تراها في الفيجوال أستوديو هي صفحة الويب،منتظرةً النص ومحتوى أداة.إذا كنت تريد البرهان،انقر زر مقطع المصدر Sourceفي الزاوية اليسارية الدنيا من العرض،أو استخدم قائمة عرض View >> قائمة الترميزMarkup menu. تتغير النافذة إلى الكود المصدري لـ HTML.

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Default.aspx.vb" Inherits="_Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
    </div>
    </form>
</body>
</html>
```

حسناً معظمه HTML.يوجد سطر في الأعلى يبدأ بـ@%> وهو لايبدو مشابه لـ HTML حقيقي- وهو كذلك،إنه تعليمات صفحة آسبي دوت نت.وهي تتضمن خاصيات تساعد على توجيه آسبي دوت نت في معالجة الصفحة.مستعيرةً المعيار من سلف ASPالسابق،يستخدم آسبي

دوت نت زوج القوسين <%...%> لتعليم أوامر معينة لآسبي دوت نت والكود.(من المحتمل أنك تميز هذه العلامات من الفصل13،حيث أنه يتم استخدامها في محرفية XML Literals)
هذا كافي بالنسبة لـ HTML.فمن يرغب برؤيتها على أية حال؟انقر زر مقطع التصميمDesign ،أو استخدم عرض View >>المصمم Designer للعودة إلى الصفحة الفارغة. دعنا نعمل على إنشاء تطبيق يضرب عددين مزودين من قبل المستخدم مع بعضهما ونعرض الناتج.من أجل هذه الميزة البسيطة،يمكن أن نكتب فقط بعض JavaScript ونضمنه كصيغة على جهة العميل،ولكننا نحاول تجنب عمل أشياء مثل هذه.اكتب التالي في صفحة الويب:

To multiply two values together, enter them in the text fields, and click the **Multiply** button

لقد جعلت الكلمة *Multiply* بخط غامق باستخدام Ctrl-B،كما أفعل مع معالج الورد.اضغط إنتر مرة Enterواحدة.بشكل افتراضي،تخطط صفحة الويب جميع العناصر مثل مستند معالجة الورد،باستخدام طريقة تدعى "نمط التخطيط السياق*flow layout mode* ".تستطيع أيضاً استخدام الحط(التموضع)المطلق *absolute positioning* للعناصر المستقلة لوضعها عند موقع معين على الصفحة.توجد طريقة أخرى لتنظيم العناصر على الصفحة:من خلال جدول HTML،لنعمل على إضافة واحد الآن.استخدم جدول Table >> إدراج جدولInsert Table .عند يظهر حوار إدراج جدول،عين جدول مخصص مكون من ثلاث صفوف وعمودين.ومن ثم انقر موافق.سيظهر الجدول بشكل مباشر في الجسم bodyلصفحة الويب. اكتب في الخلية العلوية اليسارية **Operand 1** واكتب في الخلية التي تحتها **Operand 2** ستبدو صفحتك كما هو مبين في الشكل التالي:

لضرب قيمتين مع بعضهما أدخلهما في حقول نصية ومن ثم انقر على الزر"ضرب".

operand 1
operand 2

حتى الآن لم نعمل أكثر مما نستطيع عمله في المفكرة.ولكن الآن نحن الآن جاهزون لإضافة بعض الأدوات.إذا فتحت صندوق الأدواتtoolbox ، سترى أن الأدوات مشابهة كثيراً لما هو موجود في صندوق أدوات تطبيق نماذج ويندوز.
تم تجميع الأدوات بواسطة التخصص الوظيفي:

القياسي.*Standard*

ستستخدم الأدوات الموجودة في هذا المقطع بشكل عام لبناء واجهة المستخدم لصفحة الويب خاصتك.العديد من هذه الأدوات تمثل أدوات ويندوز قياسية بالتوافق المباشر كما في عالم نماذج ويندوز.على سبيل المثال المدخلة صندوق قائمة ListBox تنفذ أداة صندوق قائمة ListBox ويندوز قياسية ضمن صفحة ويب.بالنسبة لك،كمبرمج،تبدو هذه الأداة مثل فئات أداة الدوت نت القياسية،مع الخاصيات،الطرق،والأحداث.بالنسبة للمستخدم النهائي،فإنها تبدو مثل أداة صفحة ويب قياسية،تم تسليمها باستخدام HTML العادي.
بعض من هذه الأدوات أدوات مركبة،والتي تستخدم أدوات مبنية من عدة أدوات تعمل مع بعضها.من المكن مع صيغة جهة العميل عمل بعض هذا العمل.

البيانات.*Data*

أدوات البيانات تعالج تفاعلات تحزيم قاعدة البيانات.كما تتذكر من الفصول السابقة،لست معجب بربط الأدوات في التطبيقات القياسية.ولكن عند تربط بيانات ساكنة(ستاتيكية)من خلال صفحة ويب،فإنها تتحول بالفعل لأن تكون صيانة عظيمة للوقت.بعض هذه الأدوات تعمل على تحزيم البيانات،بينما بعضها الأخر يعمل على جلب(تقديم)البيانات الفعلية.

التحقق.*Validation*

إن المستخدمين بغاية المرح،وخاصةً عندما يعملون على إدخال بيانات غريبة ضمن برنامج النوعي.التحقق من البيانات التي يعملون على إدخالها صعب إلى حدٍ كبير في تطبيقات سطح المكتب،ولكن حتى إنها أثقل عند يتحدث نظام العميل مع التطبيق الذي يستضيف للكثير من الثواني في الساعة.تعمل أدوات التحقق على إزالة بعض العبء.فهي تختبر من أجل معظم الأنواع المشتركة من أخطاء إدخال البيانات،وتشعر المستخدم بالمشاكل،كل هذا بدون كود إضافي من جهتك.عندما تحتاج إلى عمل بعض منطق التحقق المخصص،تتيح لك الأداة CustomValidator إضافة هذا المنطق كمعالج حدث،أو صيغة على جهة العميل.

التنقل.*Navigation*

تتضمن هذه المجموعة عدة أدوات مصممة لمساعدة المستخدم على التنقل من صفحة إلى أخرى أو من مقطع إلى أخر ضمن موقع الويب.

الدخول.*Login*

هذه الأدوات تغلف ميزات إدارة كلمة المرور وتسجيل الدخول بحيث يستطيع المستخدم إنشاء حساب account مستخدم جديد،وتوفر كلمة مرور مثبتة الصحة(موثقة)،أو عمل إجراءات أخرى على صلة بالأمن.

أجزاء الويب.*WebParts*

هي حاويات أدوات يستطيع المستخدم ترتيبها باستخدام سحب وإسقاط ضمن صفحة الويب.هذا التصنيف من العرض يتيح للمستخدم إضفاء طابع شخصي على العرض لتحقيق متطلبات شخصية موجودة في عقله.تستطيع تسجيل حالة إعادة عرض أجزاء الويب WebPartsفي المرة القادمة التي يعود المستخدم إلى موقع أو صفحة ويب.

AJAX Extensions

المقدار القليل من الأدوات في هذا المقطع يساعد على دعم التخصص الوظيفي آجاكس لآسبي دوت نت.آجاكس (صيغة جافا سكريبت و XML الغير متزامنة. Asynchronous JavaScript and XML)وهو مجموعة من التقنيات المعتمدة على ويب يمكن أن تساعد في جعل صفحات الويب سريعة الاستجابة أكثر،وخاصةً من خلال تحديثات الصفحة الجزئيةpartial-page updates .وهي تقع خلف مجال هذا الكتاب.

التقرير.*Reporting*

ستجد هنا أداة عرض التقريرReportViewer ،إصدار الويب من تقنية التقرير التي شرحناها في الفصل21،فهي تعض تقارير باستخدام نفس ملفات التي عملت على بناءها من أجل تطبيق سطح المكتب.

HTML

هذه أدوات HTMLقياسية،مثل <textarea>،التي مازال يستخدمها مطوري صفحات الويب منذ سنين.توفر الفيجوال أستوديو لك بعض تدقيق صحة الخاصيات والمساعدة الفورية ،ولكن استخدام هذه الأدوات مشابه لكتابة وسم مطابق مباشرةً ضمن ترميز الصفحة.

دعنا نضيف عدة أدوات من المقطع القياسي لصندوق الأدوات إلى صفحة الويب.في الخلية السفلية اليسارية للجدول الذي أضفناه سابقاً،أضف أداة زر،امنحه الاسمActMultiple، وضع خاصية النص Textله إلى Multiply.أضف أداتي صندوق نص TextBoxإلى الخليتين العلويتين على العمود اليميني للجدول،سم أحدها FirstOperandوالثانية SecondOperand.

أضف أداة عنوان(لصاقةLabel)إلى خلية الزاوية السفلية اليمنى من الجدول.سمها Product وضع خاصية النص Textلها إلى 0(نعم صفر).

ألم تلاحظ كيف يتم إعداد كل خاصية لهذه الأدوات فهو لا يختلف عن ما فعلناه في تطبيق المكتبة الرئيسي.بسيط،حالياً،ستبدو صفحة الويب خاصتك كالتي تبدو في الشكل التالي.

لضرب قيمتين مع بعضهما أدخلهما في حقول نصية ومن ثم انقر على الزر"**ضرب**".

operand 1	
operand 2	
ضرب	0

ارجع بإيجاز إلى ترميز HTML markup من أجل هذه الصفحة بالنقر على زر مقطع المصدر Sourceعند أسفل الصفحة.إذا كنت على معرفة بـ HTML،ستلاحظ الوسم<table> من أجل الجدول الذي عملنا على إضافته،ولكن ستجد شيء لست على معرفة به ضمن صف الجدول الأول.

```
<table class="style1">
            <tr>
                <td>
                    <b>operand 1</b></td>
                <td>
                    <asp:TextBox ID="firstoperand" runat="server"></asp:TextBox>
                </td>
            </tr>
            <tr>
                <td>
                    <b>operand 2</b></td>
                <td>
                    <asp:TextBox ID="SecondOperand" runat="server"></asp:TextBox>
                </td>
            </tr>
            <tr>
                <td>
                    <asp:Button ID="ActMultiple" runat="server" Text="ضرب" Width="77px" />
                </td>
                <td>
                    <asp:Label ID="product" runat="server" Text="0"></asp:Label>
                </td>
            </tr>
        </table>
```

الوسم <asp:TextBox>.فهي تبدو شيء ما مثل وسوم HTML الأخرى،ولكن لا توجد وسوم لـ HTMLتبدأ بـ "asp ".وهذا وسم آسبي دوت نت خاص ويمثل فئة أداة نماذج ويب.هذه الأدوات،والمواصفات "runat="serverالمبعثرة على طول الترميزmarkup ، هي ما تجعل صفحات آسبي دوت نت ASP.NET ما هي عليه.كعمليات آسبي دوت نت تجرد الصفحة*aspx*. وسوم هذه الصفحة المخصصة وتستدعي على الأدوات ذات الصلة إنتاج مستعرض HTML الحيادي الخاص بهاbrowser-neutral HTML .

انتهت واجهة المستخدم،لنعمل على إضافة المنطق.نريد من البرنامج أن يعمل على ضرب عاملين مع بعضهما عند النقر على زر الضرب.ارجع إلى صفحة الويب وانقر نقراً مزدوجاً على زر الضرب.يقفز إلى قالب الكود من أجل حدث نقر Clickالزر،كما تتوقع منه ذلك.

```
Partial Class _Default
    Inherits System.Web.UI.Page
    Protected Sub ActMultiple_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
ActMultiple.Click
    End Sub
End Class
```

هدف تصميم آسبي دوت نت كان أن يكون لديك كود قريب إلى كود تطبيق سطح المكتب قدر الإمكان،وهذا هو.أضف المنطق التالي إلى معالج الحدث هذا:

```
Protected Sub ActMultiple_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
ActMultiple.Click
        ' ضرب العددين
        product.Text = Val(firstoperand.Text) * Val(SecondOperand.Text)
        If (Val(product.Text) < 0) Then
            product.ForeColor = Drawing.Color.Red
        Else
            product.ForeColor = Drawing.Color.Black
        End If
    End Sub
```

عندما كنا نكتب،ألم تلاحظ استجابة المساعدة المباشرة لكل كبسة مفتاح؟لا يمكن أن أقول أن هذا كان تطبيق معتمد على ويب،وهذا عظيم. اضغط المفتاح F5لبدء التطبيق.سيطلب منك تشغيل التصحيح الذي تريد عمله.وهذا سيعدل ملف التطبيق*web.config* لدعم التصحيح debugging،فيما بعد ستعمل على تعطيل هذه الميزة بحيث لن يكون المستخدمين قادرين على تصحيح التطبيق. إذا فتحت الملف *web.config*،سترى السطر التالي:

```
<compilation debug="true" strict="false" explicit="true">
```

فقط اعمل على تغير مواصفة debugإلى خطأ falseلتعطيل التصحيح.إن آسبي دوت نت هو تطبيق خادم server application، فهو يطلب الحياة،والتنفس لخادم ويب web server قبل أن يتمكن من معالجة الصفحة.من المحتمل أن يكون أو لايكون قد نصبت خادم معلومات الانترنت Internet Information Server على نظامك،ولكن لاتقلق،فالفيجوال أستوديو تتضمن "خادم تطوير آسبي دوت نت خاص بها ASP.NET Development Server" وهو موجود تماماً لذلك تستطيع اختبار تطبيقات آسبي دوت نت.يبين الشكل التالي ظهور popping up صينية النظام system tray.



يبين الشكل التالي التطبيق المشغل في مستعرض الويب الخاص بي،مستكشف الانترنت.



إذا كنت لاتحب طريقة توزيع الجدول عبر الصفحة،تستطيع تعديله باستخدام لوحة خاصيات CSS(صفحات النمط التسلسلية Cascading Style Sheets).ارجع إلى الفيجوال أستوديو،واختر قائمة عرض View >> خاصيات CSS Properties. تفتح اللوحة، عدلها نفسها بالاعتماد على عنصر الاختيار الحالي للصفحة.للتخلص من الصلة الغرامية للجدول مع الحافة اليمينية،اختر الجدول نفسه،تحرك للأسفل إلى خاصية Position/Width،و اعمل على إزالة القيمة100% من مدخلة الخاصية.

أكثر حول الأحداث.More About Events

حتى الآن،يبدو تطبيقنا مشابه لتطبيق سطح المكتب،عرضت الفورم السحب والإسقاط الأولي الذي عملناه وإعدادات الخاصيات،وهي تستجيب لنقر الزر بالعودة إلى معالجة الكمبيوتر للمنطق.ولكن لنكن صادقين.لاتوجد طريقة تمكّن تطبيق ويب أن يكون سريع الاستجابة للأحداث كما هو الحال بالنسبة لتطبيق سطح المكتب.ما الذي يحدث عندما ينقطع اتصال الانترنت أو بكل بساطة يصبح بطيء؟كيف ستعالج أشياء مثل أحداث TextChangedفي حقول النص؟فليس لديك إمكانية عودة صفحة الويب إلى خادم الويب كل مرة يضغط المستخدم على مفتاح ما. إن لأداة صندوق النص TextBoxالتابعة لآسبي دوت نت حدثTextChanged، ولكنه لاينطلق بالنسبة لكل ضربة مفتاح.في الحقيقة،لايتم إطلاقه على الإطلاق(بشكل افتراضي) حتى يسبب شيء ما (مثل نقر زرbutton click) رجوع الصفحة إلى الخادم.وتوجد الكثير من أحداث أدوات أخرى تعمل مثل هذا.فجميعها تنتظر المعالجة حتى يقوم بالمستخدم بعمل شيء ما يعود بكامل الصفحة إلى خادم ويب. فقط عند ذلك الوقت يتم إطلاق الأحداث المؤجلة،وتتواصل المعالجة كالعادة.لذلك يوجد حقاً نوعين من الأحداث:النظامية regularوالراقية أو الاستثنائية premium. أعني المرجعية postbackوالغير مرجعيةnon-postback، الأحداث المرجعية *Postbackevents*هي تلك التي تعمل على العودة بالصفحة مباشرةً إلى المخدم من أجل المعالجة.أما الأحداث الغير مرجعية*Non-postbackevents* تؤجل معالجة الحدث حتى يعمل شيء ما أخر على عودة الصفحة إلى الخادم.معظم الأحداث إما من هذا النوع أو من ذاك،ولكن بعضها يمكن تغيره.فأداة صندوق الاختبار CheckBoxلديها الحدث CheckedChangedوالذي يتم إطلاقه بطريقة غير مرجعية non-postback عندما يعمل المستخدم على تبديل حالة صندوق الاختبار.مهما يكن،إذا وضعت خاصية AutoPostBackهذه الأداة إلى صحTrue، فإن الصفحة ستعود مباشرةً إلى الخادم في أي وقت ينقر المستخدم على صندوق الاختبار.بالإضافة لأحداث الأدوات،كامل الصفحة لديها عدة أحداث.والأكثر أهمية هو حدث تحميل الصفحة Page_Load.وهو مناظر لحدث تحميل الفورم Form_Load من أجل نماذج ويندوز،وهو مكان عظيم لتركيب خاصيات الأدوات الأولية (التمهيدية)، ملئ قوائم السياق(القوائم المنسدلةdrop-down lists)،وهكذا.سأعمل على إضافة الكود التالي إلى حدث تحميل الصفحةpage's Load .

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        ' التمهيد للبيانات
        product.Text = "فارغ"
    End Sub
```

والآن ستعرض لصاقة Product "فارغ"في المرة الأولى التي تظهر فيها الصفحة.الشيء الوحيد بخصوص معالج حدث التحميل هو أنه يتم تشغيله مرة واحدة كلما تم عرض الصفحة.بما أن تطبيق الاختبار هذا يحافظ على استخدام نفس الصفحة مرة بعد مرة من أجل نتائجها،فإن

معالج حدث التحميل سيشتغل من جديد كل مرة.من أجل برنامج الاختبار هذا،فليس له أهمية،يعمل الكود في معالج حدث نقر الزر على إعادة قيادة overrides قيمته التمهيدية"فارغ".ولكن في تطبيقات أخرى،من المحتمل أنك لاتريد الاحتفاظ بالإسناد التمهيدي للبيانات. لحسن الحظ،سيتيح لك الحدث تحميل Loadمعرفة إذا هذه هي المرة الأولى،من خلال عضو مستوى الصفحة الذي يدعى IsPostBack.

```
'التمهيد للبيانات، ولكن فقط للمرة الأولى
    If (Me.IsPostBack = False) Then product.Text = "فارغ"
```

الحالة وحالة العرض.State and View State

إذا كنت أعيد تحميل الصفحة من جذورها كل مرة مع الحاجة إلى القيم التمهيدية في معالج حدث تحميل الصفحة،كيف احتفظ صندوقي النص بالقيم المدخلة من قبل المستخدم عند إعادة تحميل الصفحة؟فنحن لم نعمل على إضافة أي كود لحفظ وتجديد تلك القيم خلال التمهيد. إليك القصة.على الرغم من أن حدث تحميل الصفحة Page_Load يعمل على منحك فرصة تمهيد الصفحة كل مرة يتم تحميل الصفحة،فمن أجل معظم الحقول ستتذكر الصفحة ما كانت عليه الحقول.تذكر أنه تم تصميم آسبي دوت نت بحيث تعتقد أنه يعمل مثل تطبيق نماذج ويندوز تماماً.ولن تكون سعيداً إذا ما فقدت الحقول على نموذج سطح مكتبك قيمها كلما نقر المستخدم زر.ولن تكون سعيداً إذا نظف تطبيق نماذج ويب هذه الحقول أيضاً.لأن صفحات الويب تكون منفصلة عن خادم الويب معظم الوقت.فتحتاج صفحة الويب طريقة ما لتحتفظ بالحالة- إعدادات البيانات والخاصية الحالية لكل أداة بين تحميلات الصفحة.يقوم نظام نماذج ويب بعمل هذا من خلال ميزة تدعى"حالة العرض".إليك كيف تعمل:تتضمن كل صفحة ويب لآسبي دوت نت حقل ستاتيكي مخفي يحتوي نسخة متسلسلة لجميع معلومات الحالة الهامة للأدوات.عندما يعمل المستخدم تغيرات إلى كل أداة ويطلق حدث ما يعمل على إعادة الصفحة إلى خادم ويب،فإنه يعمل على إعادتها مع حالتي العرض المضمنتين(مجمعة من التركيب السابق للصفحة)وجميع الإعدادات الحالية لكل أداة.باستخدام المعلومات المجمعة هذه،يكون آسبي دوت نت قادر على إعادة بناء حالة العميل المرئية الصحيحة لكل أداة،وينقل ذلك بشكل صحيح لك في كود معالج حدث جهة الخادم .
عندما تشغل تطبيق آسبي دوت نت،استخدم القائمة عرض View >> المصدر Sourceفي مستكشف الانترنت أو في مستعرضك المفضل،وسترى شيء مشابه للتالي:

```
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUKMTEyMTc3MTQwNg9kFgICAw9kFgICBw8P
FgIeBFRleHQFB05vIERhdGFkZGQME+xLedutk85TvXy9OJd
kQF02YA==" />
```

هذا هي حالة العرض.لا تسألني كيف تعمل،فلن أخبرك(لأنني لا أعلم).ولكن ليس من الأهمية معرفة كيف تعمل.ما هو هام معرفة أن آسبي دوت نت يعرف كيف يعمل مثل هذا بحيث يستطيع حفظ تطبيقك يعمل مثل نظام نماذج ويندوز.
عندما تضيف أدوات إلى صفحتك ،تعمل حالة العرض View Stateعلى الزيادة في الحجم.بما أن جميع محتوى صفحة ويب يجب أن يتم نقله بشكل متكرر إلى الانترنت،وحالة العرض الأضخم تنتج في وقت انتقال أطول.من الممكن تعطيل حالة العرض من أجل أدوات معينة باستخدام خاصيتها EnableViewState.إذا كنت لاتريد الاحتفاظ بقيمة أداة من استخدام صفحة إلى استخدام صفحة،فمن الجيد تعطيلها.

التحقق من البيانات.Data Validation

لأن هذا كود بسيط يستخدم دالة لفيجوال بيسكVal لمعالجة البيانات المزودة من قبل المستخدم،فعلى الأغلب تعمل دائماً بدون أخطاء.أي بيانات تعتبر غير صحيحة يتم تحويلها ببساطة إلى صفر.يوجد خيار أخر لمعاقبة المستخدم من أجل المدخلات الغير صحيحة قبل أن تحدث المعالجة، لتقيم *validate* القيم المزودة.المدققات validatorsالخمس في مقطع التحقق Validationلصندوق أدوات toolboxنماذج ويب يساعدك على عمل التالي فقط:

- تؤكد الـ RequiredFieldValidatorعلى المستخدم أن يزود أي قيمة في أداة ما.
- يتذمر الـ RangeValidatorإذا لم تقع قيمة أداة بين قيمتين.
- تتيح لك الـ RegularExpressionValidatorمقارنة قيمة أداة على نموذج تعبير نظامي.على سبيل المثال،تستطيع مقارنة إدخال المستخدم للرقم المتسلسل إلى نموذج(عينة) لضمان أنه يحتوي حرفين متبوعين بخمس أرقام.
- تتضمن الـ CompareValidatorأداتين،تقارن القيمة بينهما.تستخدم الأداة أيضاً كـمقيم نوع بيانات validator، مؤكدةً على أنها حقل مفرد يحتوي على نوع مناسب من البيانات،مثل قيمة بيانات أو عدد صحيح انتغر integer .
- تتيح لك الـ CustomValidatorعمل أي نوع من التحقق الذي تريده من خلال كود توفره أنت.

جميع هذه الأدوات المذكورة في الأعلى تعمل تحقق من جهة الخادمserver-side validation، وبشكل اختياري تعمل تدقيق بياناتها باستخدام صيغ جهة العميل client-side scripts(الافتراضي).فاحتواء التدقيق في جهة العميل يعمل على تقليل الحاجة للعودة إلى خادم الويب لضمان أن الحقل المطلوب لديه بيانات.واحتواء تفحص (تدقيق)على جهة الخادم server-side check يضمن أن البيانات محققة أو صحيحة حتى ولو عمل العميل على تعطيل دعم الصياغة scripting .
تعرض المتفحصات(أو المدققات validators)رسائل الخطأ الخاصة بها،لذلك تعمل على وضعهم على الصفحة متى أردت أن تظِهر رسالة الخطأ.تستطيع أيضاً أن تمتلك عدة مدققات تعرض قضايا تراكمية في موقع واحد باستخدام الأداة ValidationSummary. دعنا نضيف بعض التحقق إلى حقلي الإدخال في مثال الضرب.نريد أن نضمن أن البيانات الموفرة،وكلا القيمتين هي انتغر صحيحة.لعمل هذا،يجب أن نضيف كل من الأداة RequiredFieldValidatorوالأداة CompareValidatorلكل حقل.انقر يمين أسفل الخلية اليمينية من الجدول،تماماً بعد لصاقة label الناتج Product،واختر إدخال Insert>>عمود إلى اليمين Column to the Right من القائمة المنسدلة التي تظهر.في الخلية الجديدة اليمينية العلوية،أضف الأداة RequiredFieldValidator.ضع الخاصيات التالية لها:

ضع ControlToValidateإلى FirstOperand.
ضع Displayإلى Dynamic.يتيح هذا لحجم المدقق بأن يتقلص إلى لاشيء عندما لايكون هناك خطأ سيتم عرضه.
ضع ErrorMessageإلى Missing.

إلى اليمين تماماً من المدققvalidator ، في نفس خلية الجدول،أضف CompareValidator وضع الخاصيات التالية لها:

ضع ControlToValidateإلى FirstOperand.
ضع Displayإلى Dynamic.
ضع ErrorMessageإلى Must be an integer.
ضع Operatorإلى DataTypeCheck.
ضع Typeإلى Integer.

أضف زوج مشابه من الأداتين السابقتين إلى الصف الثاني من الجدول،استخدم SecondOperand من أجل ControlToValidate.ستبدو صفحة الويب مشابهة للشكل التالي.
شغل البرنامج وحاول إدخال بيانات خاطئة في خلايا الإدخال.ستغتاظ(تعترض)الصفحة مباشرة عندما تنقر على الزر "ضرب".هذا كل ما نستطيع عمله الآن.حفظت نسخة من المشروع هذا لك بالدليل الجزئي التابع لمشاريع الفصول في ملف المشروع لهذا الفصل.

لضرب قيمتين مع بعضهما أدخلهما في حقول نصية ومن ثم انقر على الزر"ضرب".

operand 1		MissingMust be integer
operand 2		MissingMust be integer
ضرب	0	

تكامل قاعدة البيانات.Database Integration

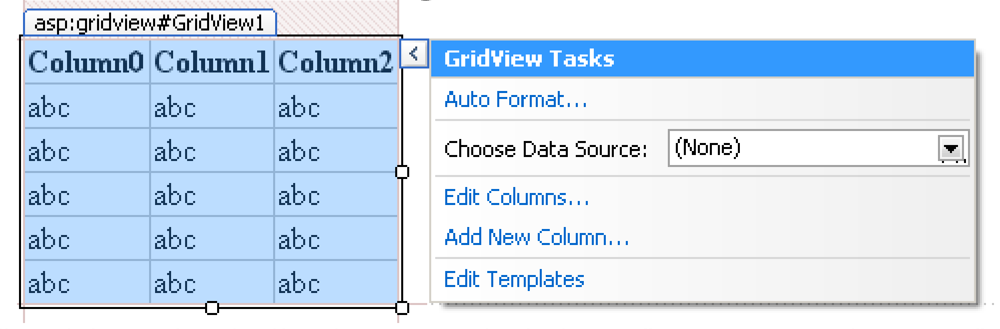
اتصال صفحات آسبي دوت نت إلى قاعدة البيانات،وخاصة إذا كنت تستخدم بعض ميزات نمط المعالج السحري wizardstyle للفيجوال أستوديو،سهل جداً جداً.وهذا لأن العديد من الأدوات المضمنة مع آسبي دوت نت تم تصميمها بشكل خاص لغرض التفاعل مع البيانات من مصدر بيانات جدولي(مستوي tabular).سنختبر(أو نضع تحت الاختبار try out) مثال معالج سحري سريع هنا،ونعمل أكثر من تكامل قاعدة البيانات في مقطع مشروع هذا الفصل.في الفصل20،تعرض التقارير الخمسة الأولى الجاهزة(المبنية داخلياً) التي عملنا على إنشاءها لمشروع المكتبة قائمة بجميع البنود المستخرجة checked-out. وصممنا تقرير RDLC لها،وبما أن ASP.NET يتضمن أداة عرض تقرير RDLC،نستطيع إعادة استخدامها من أجل تقرير معتمد على ويب.ولكن بالمقابل سنعرض التقرير باستخدام واحدة من أدوات نماذج ويب،وهي GridView.إليك الاستعلام الذي يستخلص البنود المستخرجة:

```
"SELECT PA.LastName + ', ' + PA.FirstName AS PatronName,PA.Barcode AS PatronBarcode,PC.DueDate,
IC.CopyNumber, IC.Barcode AS ItemBarcode,NI.Title, CMT.FullName AS MediaName FROM Patron AS PA INNER
JOIN PatronCopy AS PC ON PA.ID = PC.Patron INNER JOIN ItemCopy AS IC ON PC.ItemCopy = IC.ID INNER JOIN
NamedItem AS NI ON IC.ItemID = NI.ID INNER JOIN CodeMediaType AS CMT ON NI.MediaType = CMT.ID
WHERE(PC.Returned = 0)AND PC.Missing = 0 AND IC.Missing = 0 ORDER BY NI.Title, IC.CopyNumber,
PA.LastName, PA.FirstName"
```

إن هذا يجب أن يبدو مألوفاً.أنشئ موقع ويب لأسبي دوت نت جديد new ASP.NET web site من خلال الفيجوال أستوديو.اكتب السطر التالي عند أعلى صفحة المحتوى:

ACME Library Checked Out Items

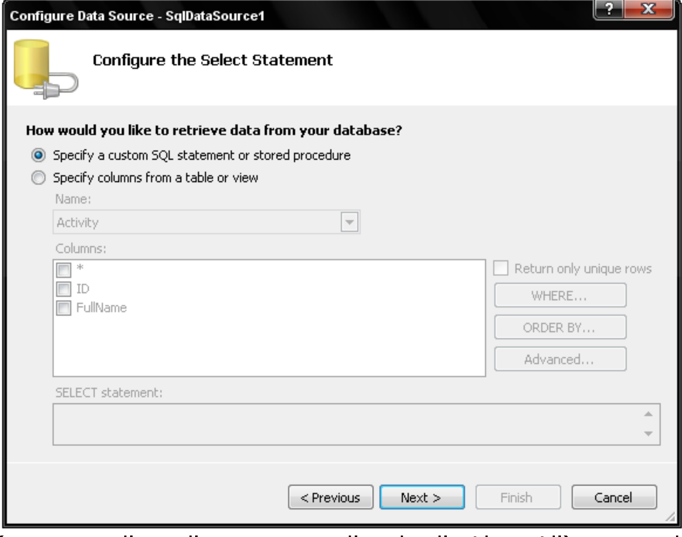
وأنت حر في تزينها بحيث تبدو أجمل.أنا عملت على تضمين الوسوم<h1> حولها في الترميز لجعلها قائمة بذاتها.تحت سطر العنوان ذاك،أضف أداة GridViewإلى الصفحة.لقد وجدتها في مقطع البيانات Dataلصندوق أدوات الفيجوال أستوديو Toolbox الخاص بي. الوسم السريع للأداة يفتح وتظهر لوحة من مهمات GridView،كما هو مبين في الشكل التالي.



إذا أردت النقر على المهمة "تنسيق تلقائي Auto Format" وتغيير المظهر للشبكة،تستطيع ذلك،ولكن المهمة الهامة حالياً هي اختيار مصدر البيانات Choose Data Source. اختر من القائمة <Add New Source>.فيظهر المعالج السحري لتركيب مصدر البيانات،مع بعض التغيرات النوعية لأسبي دوت نت.اختر قاعدة البيانات من أجل مصدر البيانات وانقر موافق OK. عندما يطلب منك الاتصال،سيكون لديك سابقاً اتصال قاعدة بيانات المكتبة في القائمة.اخترها(أو اعمل على إنشاء اتصال جديد)وانقر التالي Next .
سيطلب منك حفظ نص الاتصال في ملف تركيب التطبيق.إذا عملت على الحفظ،سيتم إضافة مدخلة إلى مقطع <connectionStrings> لملف *web.config*المنشئ لتطبيق ASP.NET.إذا كنت تريد تشغيل دعم الألعاب مع مدير نظامك،اترك الحقل غير مختار.ولكن إذا كنت تريد طريقة سهلة لتعديل معلومات الاتصال فيما بعد،فالأفضل ترك الحقل كما هو(مختار كما هو مبين في الشكل التالي).



امنح المدخلة اسم مناسب.ومن ثم انقر التالي.يطلب منك المعالج السحري من أجل الجدول وتفاصيل الحقل.اختر"تعين عبارة سكول أو الإجراء المخزنSpecify a custom SQL statement or stored procedure"، انقر التاليNext، كما هو مبين في الشكل التالي.



انقر التالي،واكتب عبارة استعلام البند المستخرج(الاستعلام السابق الذي يستخرج البنود المستخرجة) في حقل عبارة سكول SQL statement كما هو مبين في الشكل التالي.

Configure Data Source - SqlDataSource1

Define Custom Statements or Stored Procedures

Click a tab to create a SQL statement for that operation.

SELECT | UPDATE | INSERT | DELETE

SQL statement:

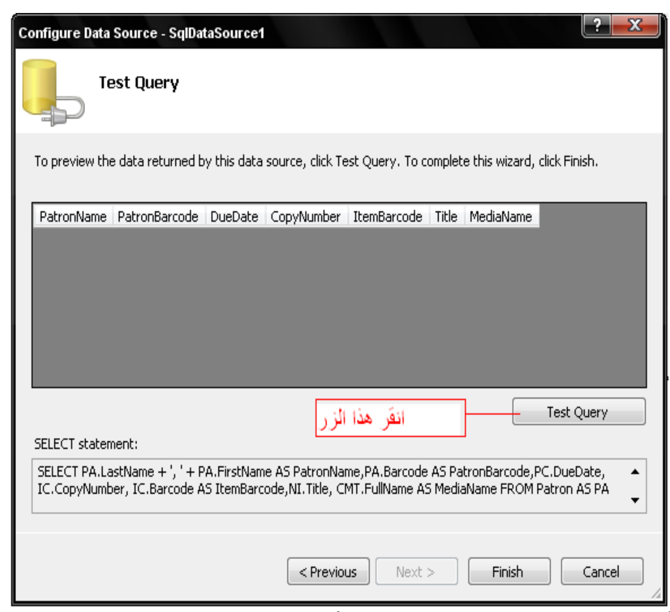
```
SELECT PA.LastName + ', ' + PA.FirstName AS PatronName,PA.Barcode AS
PatronBarcode,PC.DueDate, IC.CopyNumber, IC.Barcode AS ItemBarcode,NI.Title,
CMT.FullName AS MediaName FROM Patron AS PA INNER JOIN PatronCopy AS PC ON PA.ID =
PC.Patron INNER JOIN ItemCopy AS IC ON PC.ItemCopy = IC.ID INNER JOIN NamedItem AS NI
ON IC.ItemID = NI.ID INNER JOIN CodeMediaType AS CMT ON NI.MediaType = CMT.ID
WHERE(PC.Returned = 0)AND PC.Missing = 0 AND IC.Missing = 0 ORDER BY NI.Title,
```

Query Builder...

Stored procedure:

< Previous | Next > | Finish | Cancel

انقر التالي مرة ً أخرى.يعطيك المعالج السحري فرصة أخيرة لاختبار الاستعلام قبل أن تنقر على الزر"إنهاء Finish " كما هو مبين في الشكل التالي.



والآن هنا الجزء الأبسط.تتصل الفيجوال أستوديو إلى قاعدة البيانات،تقرأ المخطط والأعمدة المنشئة في الشبكة المصممة بإتقان من أجل الاستعلام.اكتمل تطبيقك اضغط F5 لتشغيل التطبيق.سنتوقف هنا الآن،واحفظ هذا من أجل مشروع المكتبة في مقطع المشروع لهذا الفصل.

مؤسس اتصال ويندوز.Windows Communication Foundation

هل حاولت استخراج قطعة من البيانات من موقع ويب لتستخدمه في تطبيق الفيجوال بيسك؟إذا كان الجواب لا؟لذلك دعني أخبرك:إنه "كشط الشاشة أو تنظيف الشاشةscreen scraping "،وهو مزعج.معظم مواقع الويب مع محتواه الثمين تم تصميمها من قبل ناس أنانيين،مبرمجين ما يهمهم فقط احتياجات شركتهم الخاصة ولا شيء أخر بالنسبة للمطورين الآخرين الذين يريدون سرقة البيانات الأساسية من!،أعني،الذين يريدون إضافة قيمة إلى تطبيقاتهم الخاصة لتحسين محتواها من مشارك أخر موثوق.

قشط الشاشة عامة شيء سيء.وليس فقط محتوى HTMLغريب وصعب التفسير،ولكن لن تعلم أبداً متى سيعمل مالك الموقع على تغيير المحتوى دون أن يتصل بك أولاً وبلا مجاملة.لحسن الحظ،يوفر مؤسس اتصال ويندوز حل لهذه المشكلة.تقنية ميكروسوفت الجوهرية هذه،كانت سابقاً(في الزمن السابق) كود مسمىIndigo ،يتواجد لنقل البيانات الهامة بين التطبيق والنظام،محلي localأو بعيدremote. يتم اختصار مؤسس اتصال ويندوز عادة كـ WCF.يربط عدة تقنيات مميزة إلى كلٍ موحدunified whole: طوابير الرسائل message queues (مثل MSMQ)،خادمات ويب(لاحظ الملاحظة القادمة)،المداولات الموزعةdistributed transactions (مثل MSDTC)،و تحكم دوت نت البعيد .NET Remoting.بما أن كل من هذه التقنيات تتضمن نقل المعلومات من تطبيق لأخر،فكان لايستحق العناء من ميكروسوفت أن تنفق نقوداً أكثر مما قد رأيته على الخدمة المدمجة.

إذا ما كان موقع ما لديه محتوى أو عمليات تحتاج أن تُستخدَم من قبل تطبيقات خارجية،فيمكن أن تتضمن "خدمةservice " على الموقع تجعل كشط الشاشة غير ضروري.ينفذ WCFمكافئ استدعاءات دالة معتمد على تبادل الأنظمة،يكتمل بالوسيطات والقيم المعادة،وكل ذلك يمكن الوصول له عن بعد.وكل هذا يعتمد على معايير النشر مثل بروتوكول إمكانية الوصول لكائن بسيط Simple Object Access Protocol (SOAP) والذي يستخدم نص بسيط و XML لمحاكاة استدعاء دالة بين نظامين.

ملاحظة:

قبل الفيجوال بيسك2008، استخدموا مطورو الدوت نت نظام يدعى خدمات XMLلويب XML Web Services الذي يوفر تخصص وظيفي مشابه لقسم خدمات WCF.إذا كنت تريد استخدام خدمات XMLلويب،فهي ما تزال متاحة في الفيجوال بيسك2008.مهما يكن،توصي ميكروسوفت باستخدام البرنامج الجديد WCF بدلاً من خدمات XMLلويبXML Web Services .

يتم تضمين الكثير من التقنيات في WCFلجعله ممكن،ولكن عملياً لا تحتاج إلى معرفة هذه التقنيات،بالمقابل،ستنفذ واحدة أو أكثر من طرق الفيجوال بيسك المعتمدة على واجهة تحددها أنت.وهذه الواجهةinterface ،يتم ترميزها بمواصفات WCF معينة،تعمل على تأسيس "عقد الخدمةservice contract" الذي يجعل الاتصال الوظيفي بين نظامين ممكن.تظهر خدمات WCF كملفات *.svc* على موقع الويب الخاص بك.في الفيجوال أستوديو،تستطيع إما إنشاء موقع ويب جديد new web site واختيار خدمة WCF كنوع مشروع،أو إضافة بند خدمة WCF Service إلى مشروع موقع ويب موجود.عندما تعمل ذلك،تضيف الفيجوال أستوديو الملفات الضرورية لمشروعك.الاختيار الأول هو ملف *.svc* حقيقي.فهو قناة واجهة سريعة smart interface بين موقع الويب وكود خدمة الويب الحقيقي.إليك ما وجدته في ملف الخدمة *.svc* الخاص بي.

```
<%@ ServiceHost Language="VB" Debug="true" Service="Service" CodeBehind="~/App_Code/Service.vb" %>
```

تشير هذا التوجيهات إلى مستدعي فئة الخدمة Serviceفي ملف الكود المصدري *Service.vb* المرافق.ذلك الملف هو أكثر أهمية.إليك قسم من ذلك الملف:

```
Public Class Service
    Implements IService
    Public Sub New()
    End Sub
    Public Function GetData(ByVal value As Integer) As String Implements IService.GetData
        Return String.Format("You entered: {0}", value)
    End Function
    Public Function GetDataUsingDataContract(ByVal composite As CompositeType) As CompositeType
Implements IService.GetDataUsingDataContract
        If composite.BoolValue Then
            composite.StringValue = (composite.StringValue & "Suffix")
        End If
        Return composite
```

```
    End Function
End Class
```

يبدو هذا الكود بسيط بالنسبة لي،وستعمل على تبديله عندما تكتب الخدمة الخاصة بك.تنفذ فئة الخدمة Serviceفي الكود أعضاء الواجهة IService الموجودة في ملف *IService.vb*ذو الصلة .

```
<ServiceContract()> _
Public Interface IService
    <OperationContract()> _
    Function GetData(ByVal value As Integer) As String
    <OperationContract()> _
    Function GetDataUsingDataContract(ByVal composite As CompositeType) As CompositeType
    ' TODO: Add your service operations here
End Interface
```

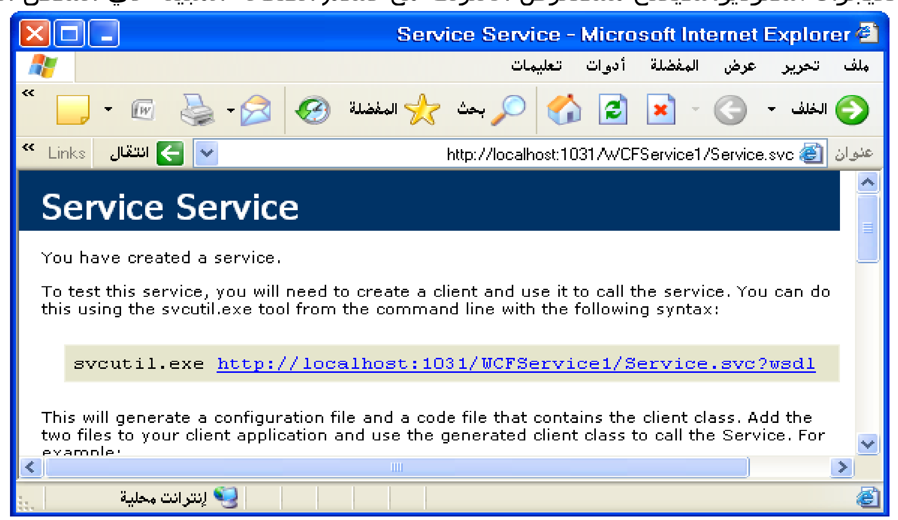
كما هو مبين،تحتوي الواجهة أيضاً عضو ليس له أهمية، GetData والذي يجب أن يتم تبديله،وتم ترميزه من خلال مواصفة <OperationContract> المرتكزة على WCF،والتي تصرح،على طول مع مواصفة الواجهة<ServiceContract>، "توجد خدمة WCFهنا"،تذكر أن المواصفة تضيف توصيف بيانات إلى مجمع ما لذلك إن المترجم أو بعض البرامج الأخرى ستعمل شيء خاص مع البنود المرمزة .في هذه الحالة،،تخبر WCF المواصفة<OperationContract>على الطريقة GetData (عند التنفيذ)كعضو خدمة.يستجيب WCF بوصل(ربط) كل كود السبر(الكود المتفحص) الذي يجعل الخدمة ممكنة.سأعمل على تبديل الدالة GetDataبواحدة أخرى تدّعي على الأقل عمل شيء ما حقيقي.أولاً،سأعمل على تغير الواجهة IServiceبحيث تُعرف العقدcontract .

```
<ServiceContract()> _
Public Interface IService
    <OperationContract()> _
    Function NumberToText(ByVal sourceNumber As Integer) As String
End Interface
```

ومن ثم في الفئةService، سأنفذ الواجهة وعضوها NumberToText.

```
Public Class Service
        Implements IService
        Public Function NumberToText(ByVal sourceNumber As Integer) _
        As String Implements IService.NumberToText
            Select Case sourceNumber
                Case 0 : Return "Zero"
                Case 1 : Return "One"
                Case 2 : Return "Two"
                Case 3 : Return "Three"
                Case 4 : Return "Four"
                Case 5 : Return "Five"
                Case 6 : Return "Six"
                Case 7 : Return "Seven"
                Case 8 : Return "Eight"
                Case 9 : Return "Nine"
                Case Else : Return "Out of range"
            End Select
        End Function
    End Class
```

إذا شغلت هذا التطبيق في الفيجوال أستوديو،سيفتح مستعرض الانترنت مع قسم الصفحة المبينة في الشكل التالي.



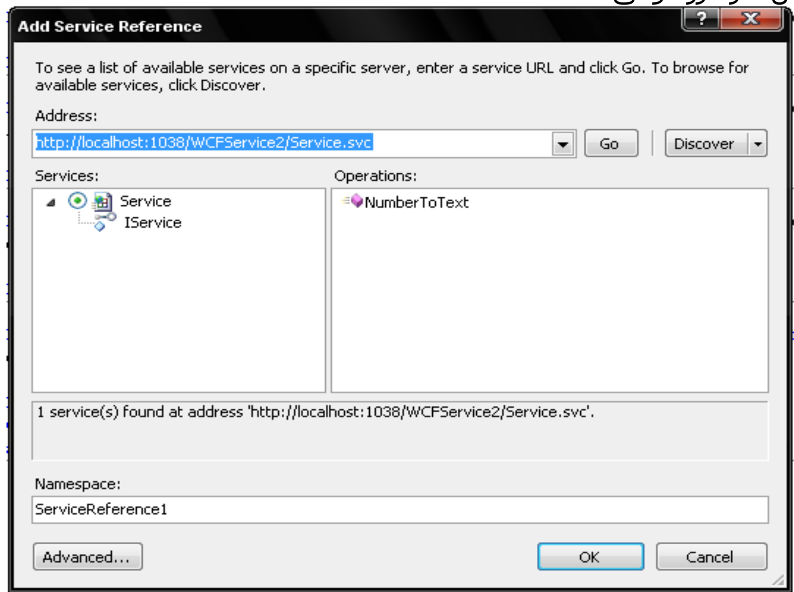
خدماتWCF هي طرق،ومستعرض الويب ليس الوسيلة النموذجية لتشغيل الإجراءات والدوال،لذلك تظهر الصفحة في الشكل السابق بدلاً عن ذلك. وهي عبارة عن محتوى إعلامي يبين لك كيف تستطيع اختبار أو استخدام الخدمة،إما من خلال فائدة(منفعة) المصممة لذلك الهدف،أو من خلال كود الفيجوال بيسك أو كود سي شارب#C .

بما أن هذه الخدمة تعمل على نظامي باستخدام خادم ويب لأسبي دوت نت،سأكتب تطبيق سطح مكتب لاستدعاء الطريقةNumberToText .
ابدأ تطبيق جديد منفصل للفيجوال أستوديو وأنشئ مشروع نماذج ويندوز.اختر مشروع Project << إضافة مؤشر(مرجع)خدمة Add Service

Reference.تظهر فورم إضافة مرجع خدمة،إنها أداة تستخدمها لإيجاد خدمات WCFالمحلية وعن بعد. بإمكانك بشكل محدد طلب الخدمة إذا كنت تعلم عنوانها.لتحديدها،انقر نقراً مزدوجاً على أيقونة خادم تطوير ويب لأسبي دوت نت ASP.NET Development Web Server في صينية النظام.سيوفر لك الحقل Root URLقاعدة العنوان.على نظامي هو الآن"http://localhost:1038/WebSite2 " على الرغم من أنه وبشكل افتراضي سيغير رقم المنفذ إذا ما عدت وشغلت الخدمة.أضف إليه اسم الملف من أجل الخدمة الخاصة بك.

http://localhost:1038/WCFService2/Service.svc

أدخل هذا العنوان في حقل عنوان address نموذج مرجع الخدمة،وانقر "اذهبGo ".إذا نجح،تحدد الفورم الخدمة وتعرض عقودها contracts في حقل الخدماتServices. يبين الشكل التالي إيجاد المعاملNumberToText. تستطيع توفير وصول بالاسم للخدمة في كودك وذلك بتغير حقل فضاء الاسمNamespace. ومن نقر الزر موافقOK .



لوضع الخدمة تحت الاختبار،عملت على إضافة أداة صندوق نص TextBox وأداة زر Buttonإلى الفورم،وأضفت الكود التالي(استخدمت فضاء الاسم الافتراضي ServiceReference1):

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        MsgBox((New ServiceReference1.ServiceClient).NumberToText(CInt(Val(TextBox1.Text))))
    End Sub
```

شغل البرنامج ،اكتب رقم من 0إلى9،ومن ثم انقر الزر.مما يستدعي بشكل صحيح خدمة ويب والعودة بالاسم الإنكليزي لنسخة العدد.وستعمل وكأن الخدمة NumberToTextتشتغل على خادم ويب لبحث واحدة من الخدمات البعيد المدى.

إضافة المساعدة عبر الشبكة.Adding Online Help

إذا كان هناك شيء تعلمته خلال ال25 سنة من البرمجة،هو أن المستخدمين دائماً بحاجة إلى بعض المساعدة لتشغيل برمجيات على أنظمتهم.ويحتاج المبرمجين مساعدة أيضاً،ولكن بالعودة إلى الكمبيوترات:فكان من النادر ما تجد مستخدم خبير.إذا كتبت تطبيقات عمل موجهة للعمل والدوائر ضمن منظمات،ستجد أن المستخدمين متقنين لأعمالهم،ولكن ليس من الضروري إتقانهم استخدام الكمبيوتر.وهذا لما من الملزم عليك صنع برامجك بسيطة قدر الإمكان.

وعليك إضافة المساعدة عبر الشبكة لتطبيقات.وهذه المستندات الجاهزة تمثل موجة الدعم الأولى حاجات مستخدمي برمجياتك.وبالطبع ومن النادر ما يقرؤونها،ولكن من الجيد أن تكون قادر القول "هل تريد مراجعة المساعدة عبر الشبكة،والتي تغطي هذه القضايا بالتفصيل".

في هذا الفصل،سنناقش خيارات المساعدة عبر الشبكة المتاحة لك في الفيجوال بيسك ونركز على *HTML Help 1.x* نظام المساعدة لويندوز إكس بي XPالقياسي .

خيارات المساعدة عبر الشبكة لويندوز.Windows Online Help Options

لقد أصبحت المساعدة عبر الشبكة جزء من ويندوز بما أنه المحرر الأولي،بالعودة إلى الأيام عندما كانت التطبيقات وأنظمة التشغيل تحافظ عل مانويلات manualsمساعدة مطبوع ولا تحتاج أكثر من قرصي فلوبي.في الحقيقة إني أشتاق لتلك الأيام.كل تلك المانويلات تغيرت واستبدلت بأنظمة المساعدة عبر الشبكة وملفات HTML"اقرأني".فالآن يمكن أن تقرأ كتاب مثل هذا ولكن ستجد الكثير من المساعدة عبر الشبكة،وخاصةً في هذه الأيام مع القدرة على تضمين صفحات المساعدة عبر الشبكة بشكل ديناميكي(تنتقل تبعاً للموضوع الذي تعرضه) والمحتوى الفعال(الشجري الترتيب وغير ذلك من ترتيبات المساعدة الفعالة).

WinHelp

كان نظام المساعدة الأساسي *WinHelp*.وكان يتضمن صفحات مساعدة بتنسيق بسيط مع وصلات صفحات مباشرة من صفحة لأخرى.يضيف ملف محتويات منفصل جدول محتويات الدعم،عليك شحن ملف المحتويات *.cnt* مع ملف *.hlp* كمجموعة .ملفات المساعدة الأساسية هذه كانت(وبعدة طرق،هي كذلك)جيدة بشكل كافي من أجل معظم حاجات المستخدمين،وما تزال مدعومة بواسطة جميع إصدارات ويندوز.

كانت ملفات WinHelpيتم تصميمها باستخدام تنسيق نص غني(أو متقدم(Rich Text Format (RTF) ،وتنسيق معالج ورد المدعوم من قبل العديد من البائعين،ولهذا السبب،كان المحتوى سهل البناء،على الرغم من أن وصلات ربط الصفحات والميزات الخاصة بالمساعدة الأخرى تتطلب نص ودمج تنسيقات مختلفة،ولكن لاقى WinHelp احتياجات المستخدمين لسنوات .

HTML Help

عندما بدأ الانترنت يغزو العالم مع مقدرته على إنتاج صفحات جميلة من خلال لغة HTMLالمعتمدة على الوسوم المعروفة.قررت ميكروسوفت تحديث نظام المساعدة الخاص بها إلى نظام يستخدم توثيقات HTMLالقياسي:مساعد*HTML Help* .وكما يدل اسمه ضمنياً، *HTML Help* هو حقيقةً معتمد على HTML.أي شيء يولد HTMLيمكن أن يولد محتوى HTML Help: أدوات مصمم صفحات ويب لمشارك ثالثي،معالجات وردword processors ،تطبيقات خاصة بك،وحتى المفكرة.كما تتوقع،بعض الأدوات المصممة من قبل بعض البائعين تكون بشكل خاص موجهة لنظام المساعدة HTML Help.

إن HTMLأفضل من WinHelp،تبعاً لاعتماده على HTMLوتقنيات أخرى مناسبة.كل صفحة لملف المساعدة عبر الشبكة هو ملف/صفحة HTMLمنفصلة.

وصلات الربط Hyperlinksبصفحات مساعدة أخرى هي وصلات HTMLمباشرة. و HTML Help يوظف معظم الميزات المستخدمة في أي صفحة ويب،متضمناً صفحات النمط التسلسلية Cascading Style Sheets(CSS)و صيغ جافاJava scripting .

ملفات HTMLالمترجمة(أو المركبة) لديها الامتداد ***.chm***.،ويتضمن ملف واحد محتوى رئيسي،جدول المحتويات،وفهرس مصطلحات محدد مسبقاً.سنستخدم تقنية HTML Help لإضافة محتوى المساعدة عبر الشبكة لمشروع المكتبة.وسأترك التفاصيل لما بعد في هذا الفصل.

Microsoft Help 2

معظم التطبيقات التي تم بيعها عند كتابة هذه السطور تستخدم HTML Help،ولكن ليس جميعها،واحد من أكبر الاستثناءات هو الفيجوال أستوديو نفسها.فنظام تعليماتها(أو مساعدتها) يعتمد على *Microsoft Help 2* (ويعرف أيضاً *HTML Help 2.x*)وهو جمع أو دمج محتوى HTML و XMLإلى مجموعة تجمعات تعمل مع بعضها كواحد.إذا كنت قد نصبت النسخة الكاملة من خادم سكول SQL Server على نظامك مع الفيجوال أستوديو،فكلاهما يتشارك واجهة مساعدة مشتركة.تستطيع حتى البحث عن الصفحات في كلا التجمعين في نفس الوقت.

تجعل ميكروسوفت "مجموعة تكامل المساعدةHelp Integration Kit " متاحة من أجل المطورين الذين يرغبون بدمج محتوى خاص بهم إلى نظام Microsoft Help 2.وهذا أكثر فائدة من أجل البائعين الذين يطورون أدوات مشاركة ثانوية بحيث تتكامل مع الدوت نتNET. أو خادم سكول SQL Server. ونادراً ما يتم استخدام Microsoft Help 2من أجل برمجيات المستخدم النهائي .

منصة العمل المساعدة .Assistance Platform

يستخدم ويندوز فيستا نظام مساعدة جديد يدعى "منصة العمل المساعدة (AP) *Assistance Platform*".جميع المساعدة عبر الشبكة التي تأتي مع الفيستا تم كتابتها باستخدامAP ،ولكن ليس بالنسبة للباقي،لأن ميكروسوفت قررت عدم إصدار هيئة الملف من أجل أن يتم استخدامه من قبل باقي البائعين.

حسناً ،يوجد القليل من الشركات الكبيرة و **OEM (office for emergency management)** تستخدم AP.ولكن أنت وأنا لسنا منهم.

طرق أخرى.Other Methods

لايستخدم كل تطبيق أنظمة المساعدة المحددة من قبل ميكروسوفت.فبعض التطبيقات لا تتضمن مساعدة عبر الشبكة على الإطلاق.

صفحات القائمة بذاتها هي مجرد استقلال واحد عن ملفات ،وهي بديل قليل الحيوية من أجل التطبيقات البسيطة،أو تلك المستضافة على موقع ويب.تستطيع استخدام تنسيقات قياسية أخرى،مثل مستندات نصية أو مستندات معالجة بالورد، فقط في حال لم تكن لديك موارد إنتاج ملفات مساعدة عبر الشبكة حقيقية.

توجد ميزة في الفيجوال أستوديو تتيح لك إنتاج توثيق من تعليقات XML المضافة لكل عضو في فئتك.(لن اشرحها راجع المساعدة عبر الشبكة الخاصة بالفيجوال أستوديو ومن أجل المزيد من المعلومات راجع فقرة تعليقات XML في التعليمات التي تنصبها مع الفيجوال أستوديو).

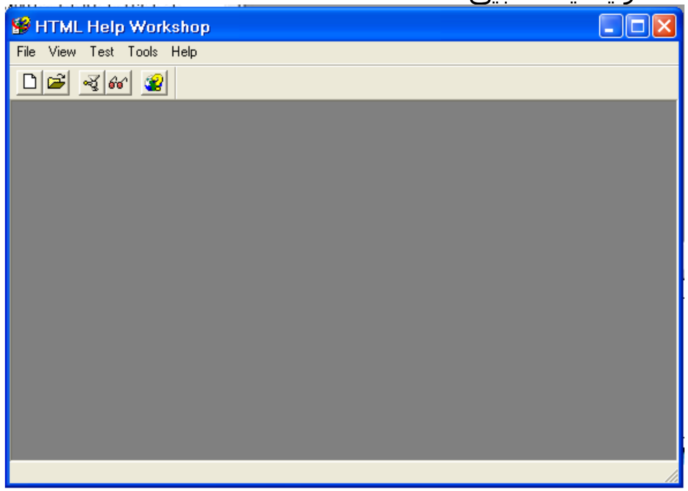
لاتفكر حتى باستخدام هذه الميزة من أجل التوثيق الموجه للمستخدم(أو التعليمات الموجهة للمستخدم) ما لم تكن تطور مكونات معتمدة على فئة من أجل أن يتم استخدامها من قبل مطورين آخرين.

تصميم HTMLالمساعد .Designing HTML Help

يتم بناء ملفات HTML المساعدةHTML Help من عدة ملفات مصدرية :

- ملفات المحتوى *Content files*،وخاصة ملفات HTML القياسية،وصل المعلومات الجوهرية للمستخدم ،إما من خلال نص ثابت ورسوميات أو من خلال سلوكيات(أو تصرفات) نمط صفحة ويب المتقدمة وصيغ متاحة بشكل طبيعي في صفحات ويب.
- ملف محتويات المساعدة *Help Contents* الذي يستخدم *.hhc* في امتداده .باستخدام وسوم HTML القياسية و ،يحدد الملف جدول هرمي(أو شجري) للمحتويات المستخدمة بواسطة ملف المساعدة .
- يستخدم ملف الكلمات المحجوزة للمساعدة *Help Keywords file* امتداد الملف *.hhk*.،ويوثق الفهرس المستخدم للوصول إلى صفحات المساعدة من خلال كلمات محجوزة خاصة محددة مسبقاً.
- ملف مشروع المساعدة،الذي يستخدم امتداد الملف *.hhp*.يعرف مشروع المساعدة الكامل وملفه *.chm*. المستهدف.وهذا ملف نصي ذو نمط أولي INI-style text يعرف جميع الملفات الأخرى التي سيتم تركيبها أو ترجمتها إلى ملف المساعدة المستهدف.ويعرف أيضاً العديد من الخيارات الواسعة للمشروع.

تستطيع بناء ملفات المحتوى الرئيسية يدوياً باستخدام أي أداة HTML قياسية تريد،طالما أن التنسيق الخارج يتطابق مع ماهو متوقع من قبل مترجم المساعدة لـ HTML(المزود من قبل ميكروسوفت).بالنسبة لملفات المحتويات،بشكل عام الأداة التي تستخدمها ليست بالقضية الهامة بما أن HTML كافي. إن أي وصلات مباشرة hyperlinks تعمل على تضمينها في المحتوى إلى صفحات مساعدة أخرى في نفس الدليل ستصبح وصلات مساعدة قياسية help links في ملف المساعدة المترجم أو المركب.
تتطلب الملفات التي لا تحتوي على محتوى تنسيق بدقة عالية،فجميعها تعتمد على HTML،ما عدا ملف مشروع المساعدة،والذي هو ملف أوليINI . ستحتاج إما إلى تصميم هذه الملفات يدوياً أو باستخدام تنسيق متوقع،أو استخدام أداة يمكنها إنتاج هذه الملفات لك في تنسيق أو هيئة صحيحة.
توفر ميكروسوفت أداة مجانية تساعدك على إنشاء ملفات بدون محتوىthe non-content files ،وضمها مع بعضها ومع ملفات المحتوى من أجل الترجمة أو التركيب النهائي.تستطيع تنزيل HTML *Help Workshop* مباشرةً من موقع ميكروسوفت على الويب .اذهب إلى مركز تنزيل ميكروسوفت في الصفحة:*http://www.microsoft.com/downloads*
وابحث عن (" HTML Help Workshop ")ستستقبل عدة نتائج،ولكن الأولى في القائمة(عند الترتيب بالأكثر شعبيةpopularity)ستكون الأداة التي تحتاجها.يبين الشكل التالي الصفحة الرئيسية لتطبيق.



في باقي هذا المقطع،سنستخدم HTML Help Workshop لبناء ملف مساعدة بسيط يحتوي صفحتين:صفحة الترحيب welcome page و صفحة "المزيد من المعلومات more information ".سأضمن هذا المشروع مع المشروع النهائي*HTMLHelpSample* .

ملفات المحتوى.Content Files

يتضمن مشروعنا الصغير ملفي محتوى: *welcome.htm* و *moreinfo.htm*.حتى ولو كانت التقنية متطورة ،عملت على صناعتها بالمفكرة.إليك محتوى الملف*welcome.htm* .

```
<html>
<head><title>مقدمة</title></head>
<body>
أهلاً بك في ملف المساعدة.للمزيد من التعليمات,
<a href="moreinfo.htm">انقر هنا</a>.
</body>
</html>
```

ملف المزيد من التعليمات مشابه كثيراً لهذا,

```
<html>
<head><title/>معلومات إضافية<title></head>
<body>
ليس هناك مايقال أكثر من ذلك،مرحباً بك
<a href="welcome.htm">انقر هنا</a>.
</body>
</html>
```

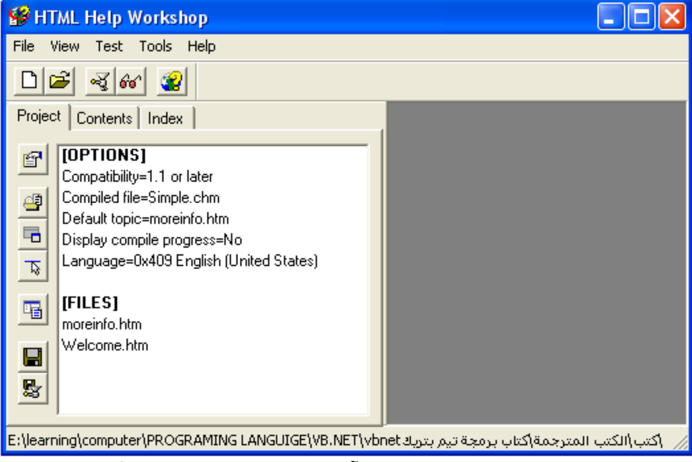
باستطاعتك إضافة ملفات صور(مثل ملفات JPEG وGIF)وتربطهم كما تعمل عادة في صفحة ويب.تأكد من تخزين ملفات الغرافيك(الصور) في نفس دليل(أو الدليل الفرعي) الملف الرئيسي من أجل سهولة الوصول.

ملف مشروع المساعدة.Help Project File

لنعمل على إنتاج باقي الملفات من خلال HTML Help Workshop.شغل هذا التطبيق،ومن القائمة ملف File>> الأمر جديد Newلإنشاء مشروع جديد.باستخدام معالج المشروع السحري،حدد موقع واسم ملف hhp.الجديد.سأعمل على إنشاء ملف مسمى Simple.hhpفي نفس المجلد حيث يوجد ملفي المحتوى.يحثك المعالج السحري من أجل ملفات عملت على إنشاءها مسبقاً.ضع إشارة صح في الحقل" HTML files "،كما هو مبين في الشكل التالي.أضف ملفي HTMLفي الخطوة التالية واعمل على إكمال المعالج السحري.يتم إنشاء ملف المشروع مع مؤشر أو مرجع إلى كل من الملفين اللذين أضفتهم.
المشروع فارغ نوعاً ما،فليس لديه حتى عنوان نافذة محددة من أجل ملف المساعدة المترجم.تستطيع وضع العنوان وإعدادات أخرى عامة من خلال خيارات المشروع،التي يتم الوصول لها من خلال الزر الأعلى على الشريط الشاقولي الذي يظهر بعد أن تنهي المعالج السحري على الجهة اليسارية من النافذة الرئيسية.تستطيع أيضاً أن تنقر نقر مزدوج على البند[OPTIONS] في قائمة تفاصيل المشروع.عندما تظهر نافذة الخيارات،ادخل "مساعدة بسيطة**Simple Help** " في حقل العنوانTitle،وانقر موافق OK.



أما ما يحويه ملف المشروع مبين في الشكل التالي:

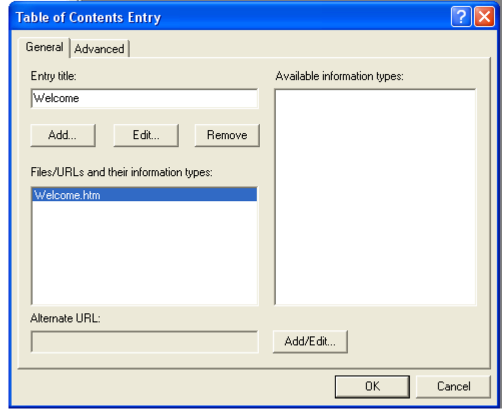


سيتغير الملف كلما أضفنا ملفين بدون محتويات two non-content files آخرين،ولكن ليس كثيراً.ترجم الملف الآن(باستخدام القائمة ملف File >> الأمر ترجم Compile) وتشغيله يعرض نافذة مساعدة بسيطة ،كما هو مبين في الشكل التالي.

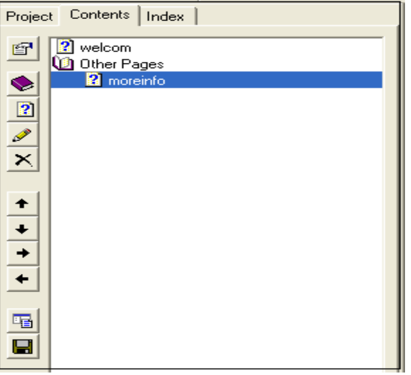
ملف محتويات المساعدة.Help Contents File

سيساعد جدول المحتويات المستخدم في قراءة واختبار المساعدة الضخمة عبر الشبكة. لإضافة ملف المحتويات،انقر على تبويب المحتويات على الجهة اليسارية Contentsمن الفورم الرئيسية،ورد على الطلب الذي ترغب به لإنشاء ملف جديد،سمه Simple.hhc.تتغير الفورم لعرض محرر جدول المحتويات. توجد طريقة أخرى لإنشاء ملف المحتويات هو باستخدام القائمة ملف File>> الأمر جديد Newواختر جدول المحتويات Table of Contents من نموذج الاختيار الجديد. وهذا أقل دليل،وكما أنه لا يرتبط مباشرةً بملف المحتويات بالمشروع.

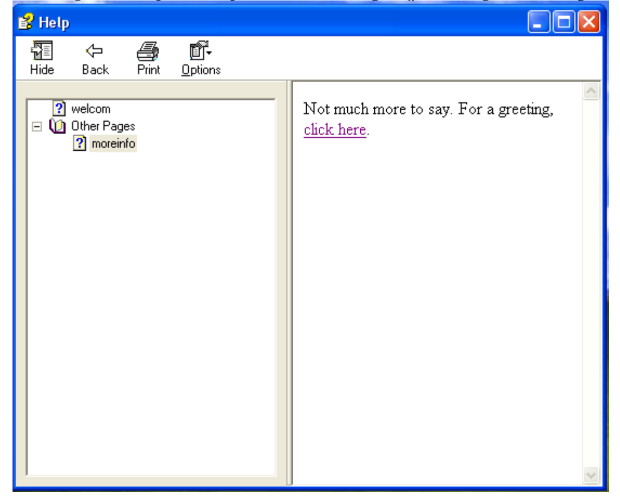
استخدم أزرار شريط الأدوات الجديد للتنقل للأعلى أو للأسفل أو اليسار أو اليمين لأضافة وتعديل مدخلات المحتوى،استخدم الزر العلوي("خاصيات المحتوياتContents Properties ")لتحرير خيارات جدول المحتويات.في نموذج خاصيات المحتويات،أزل علامة الاختيار عن الحقل"استخدم مجلدات بدل الكتبUse folders instead of books " وانقر موافق OK .
الأزرار التالية – أزرار الكتاب("أدخل رأسInsert a heading ") و زر علامة الاستفهام/الصفحة("أدخل صفحةInsert a page ")-هي الأزرار الرئيسية المستخدمة لإضافة مدخلات جديدة إلى المحتويات.قمت بالنقر على الزر" أدخل صفحةInsert a page " للحصول على جدول نموذج مدخلة المحتويات المبين في الشكل التالي:



كما هو مبين في الشكل،عملت على وضع عنوان المدخلة إلى(Welcome)،واخترت كما هو مبين في الشكل الملف(*welcome.htm*) من خلال الزر "إضافة Add ."وسأعمل المثل بالنسبة للملف " *moreinfo.htm* ."وسأعطيه العنوان " More Information ."عملت أيضاً على إضافة مدخلة رأس باستخدام زر شريط الأدوات"أدخل رأسInsert a heading " على الفورم الرئيسي،سميته"صفحات أخرىOther Pages ."استخدمت أزرار أسهم شريط الأدوات لنقل المدخلة*moreinfo.htm* إلى مقطع الرأس هذا.



إذا ترجمت وشغلت الملف،سيتضمن الآن جدول المحتويات في لوحة منفصلة،زائد شريط الأدوات.



ملف الكلمات المفتاحية(أو الفهرس).Help Keywords (Index) File

يتيح ملف الفهرس للمستخدم من الوصول إلى صفحات معينة بالبحث عن فكرة أو موضوع من قائمة.توجد علاقة عديد – إلى – العديد بين هذه الكلمات المفتاحية وصفحات المساعدة:يمكن أن تقود كلمة رئيسية واحدة إلى صفحة أو أكثر،وصفحة واحدة يمكن أن تستهدف عدة كلمات مفتاحية.

اعمل على إنشاء فهرس بالنقر على تبويب الفهرس Index على النصف اليساري من الفورم الرئيسي،وتابع طلب إنشاء ملف فهرس جديد،سمه. *Simple. hhk*. كما مع محرر المحتويات،يتضمن محرر الفهرس Indexشريط أدوات شاقولي صغير.

استخدم الزر الثاني في شريط الأدوات،الزر الذي عليه صورة أو أيقونة المفتاح،لإنشاء مدخلات الكلمات الرئيسية،سأعمل على إضافة الكلمات المفتاحية:

أولي.basic، ترتبط بالملف*welcome*.htm .

متقدم.advanced، ترتبط بالملف *moreinfo*.htm.

كل شيء.everything، ترتبط إلى كلا الصفحتين.

تعمل نموذج محرر مدخلة الفهرس تماماً مثل جدول نموذج مدخلة المحتويات،بالسماح لك بتحديد الصفحات الهدف من أجل كل كلمة مفتاحية.

احفظ وترجم المشروع مما يضيف ميزات الفهرس إلى ملف المساعدة المترجم.

تنسيق نوافذ المساعدة.Formatting Help Windows

على نظامي،تشغيل ملف المساعدة المترجم يعرض المحتوى في نافذة صغيرة في الزاوية العلوية اليسارية من الشاشة.ولكن محتوى المساعدة هام،وأريده أن يظهر أقرب إلى وسط الشاشة،وفي نافذة أكبر.لحسن الحظ،تستطيع التحكم بالنوافذ المستخدمة لعرض المحتوى.
إرجع إلى تبويب المشروع Projectوانقر على الزر الثالث على شريط الأدوات الشاقولي اليساري من النافذة.يتيح لك هذا الزر"إضف/عدل تحديدات النافذة Add/modify window definitions " تحديد واحد أو أكثر من النوافذ لأن يتم استخدامها من أجل صفحات المساعدة المختلفة في ملفك.عندما يطلب منك إضافة نافذة جديدة New Window اكتب الاسم "SimpleWindow ".
حوار أنواع النوافذ الذي يظهر يتضمن عدة خيارات للحصول على النافذة الدقيقة التي تريدها،على الرغم من ومن المحتمل أنك ستكون حريص جداً ،لنقول أن لديك 243 أنواع نافذة مختلفة.التبويب Positionهو الأكثر أهمية.فهو يتضمن زر التحجيم التلقائي Autosizer والذي يتيح لك سحب النافذة إلى الحجم المرغوب.تعديل الحجم إلى شيء تحدده،أضف إلى"نص شريط العنوانTitle bar text " النص التالي" Simple Help " على التبويب "عام General "،ومن ثم انقر موافق OK.بما أن هذه النافذة هي النافذة الوحيدة التي حددناها،فإنها ستصبح النافذة الافتراضية،وسيتم استخدامها من أجل عرض المساعدة الرئيسية في المرة التالية بعد أن تعمل على ترجمة وتشغيل الملف.

إمكانية الوصول إلى HTML Help.

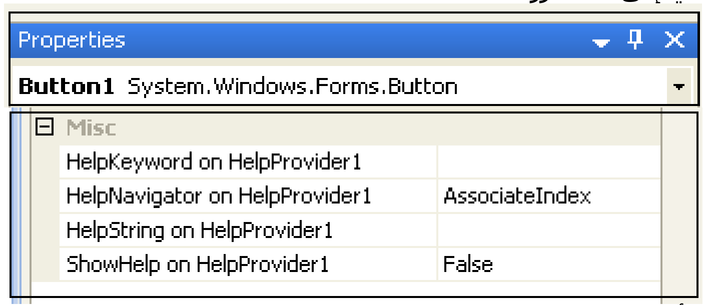
توفر الفيجوال أستوديو طريقتين رئيسيتين لتكامل المساعدة عبر الشبكة إلى تطبيق سطح المكتب.الأولى تستخدم الأداةHelpProvider ،وتوجد في مقطع المكونات Componentsمن شريط أدوات الفيجوال أستوديو .الثانية تستخدم الطريقة "أظهر المساعدةHelp.ShowHelp " لحزمة نماذج ويندوز.كلا الطريقتين تتيحان لك عرض صفحات معينة أو أجزاء ملف المساعدة ل HTML المترجم .

الأداة HelpProvider.

يمكن أن يتم إضافة الأداة HelpProviderإلى الفورم لتمكين الوصول إلى المساعدة عبر الشبكة.وهي توفر اختبار أولي للمساعدة عبر الشبكة:(1) وصول معياري إلى ملفات المساعدة HTMLالمترجمة ،(2) والمساعدة المنبثقة(أو مساعد الظهور الفوري pop-up help).كلا الطريقتين تضعان التركيز على أدوات مستقلة للفورم،وعلى ميزات مساعدة خاصة لكي يتم ربطها إلى كل أداة .

الوصول إلى ملفات HTMLالمساعدة.Accessing HTML help files

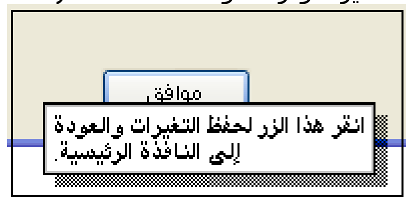
لاستخدام أداة HelpProviderمع ملفات HTMLالمساعدة المترجمة ،ضع خاصية الأداة HelpNamespaceإلى موقع ملف المساعدة الصحيح.ومن ثم اضبط خاصيات الأداة الأخرى على الفورم لتشير إلى ميزات معينة ضمن ملف المساعدة.تؤثر الأداة HelpProviderبالأدوات الأخر بإضافة العديد من الخاصيات الأخرى الإضافية إلى كل منها.يبين الشكل التالي الخاصيات الأربع الإضافية (HelpKeyword، HelpNavigator، HelpString، ShowHelp)التي تم إضافتها بشكل تلقائي إلى أداة الزر Button.



الخاصية HelpNavigatorالمضافة إلى كل أداة تعرف ميزات ملف المساعدة لإمكانية الوصول لها عندما يضغط المستخدم على المفتاح F1بينما تكون تلك الأداة قيد التفعيل.للوصول إلى صفحة معينة ضمن ملف المساعدة(مثل *welcome.htm*)،تضع الخاصية HelpNavigatorللأداة الهدف إلى Topicوتضع الخاصية HelpKeywordذات الصلة إلى اسم ملف الصفحة (welcome.htm)
الخاصية HelpNavigator من أجل أداة ما يمكن أن يتم وضعها لإمكانية الوصول إلى مقاطع غير الصفحة لملف المساعدة عبر الشبكة أيضاً. تعرض القيمة TableOfContentsمحتويات ملف عبر الشبكة،الفهرس Indexيقفز إلى فهرس الكلمة المفتاحية keyword index .توجد العديد من الخيارات الأخرى أيضاً.

إظهار المساعدة الفورية (أو المنبثقة).Showing pop-up help

تمكن أيضاً الأداة HelpProvider المساعدة "الفورية أو المنبثقة pop-up " على أدوات مستقلة.وهذا التنوع في المساعدة يؤدي إلى ظهور نافذة تلميح أداة صغيرة فوق تلك الأداة تماماً،عارضةً رسالة صغيرة توفر معلومات الاستخدام لتلك الأداة،كما هو مبين في الشكل التالي.



تعمل المساعدة المنبثقة عندما تعمل على تمكين "زر الإنبثاق"في شريط عنوان الفورم.لإعداد المساعدة المنبثقة لإداة ما،اتبع التالي:
1.أضف أداة HelpProviderإلى الفورم،ولكن لاتتعب نفسك بإعداد الخاصية HelpNamespaceإلى ملف ما.
2.ضع خاصية الفورم HelpButtonإلى صواب True.
3.ضع خاصيات الفورم MaximizeBoxوMinimizeBoxإلى خطأ False.
4.ضع الخاصية HelpString على HelpProvider1 إلى نص معلومات(كما كتبت أنا في الزر في الشكل العلوي)على كل أداة والتي ستعرض المساعدة الفورية أو المنبثقة الخاصة بها.
يعرض المستخدم المساعدة الفورية أو المنبثقة بالنقر أولاً على زر"المساعدة "ذو علامة الاستفهام في شريط عنوان الفورم ومن ثم انقر على الأداة (كما هو مبين في الشكل العلوي عند النقر على الزر "موافق").

الطريقة ShowHelp.

تعرض الطريقة System.Windows.Forms.Help.ShowHelp أجزاء معينة من ملف المساعدة HTMLالمترجم بالاعتماد على المعاملات النسبية الممررة إلى الطريقة.وهي مشابهة تماماً لجزء المساعدة المعتمد على ملف لأداة HelpProvider،ولكن ضمن طريقة تابعة للفورم.لعرض صفحة معينة ضمن ملف مساعدة ما،استخدم هذا التركيب التالي:

```
Windows.Forms.Help.ShowHelp(Me, "Simple.chm", HelpNavigator.Topic, "moreinfo.htm")
```

المعامل النسبي الأول هو المرجع أو المؤشر إلى الفورم التي تستدعي الطريقة.الطريقة الشائعة لاستخدام هذه الطريقة هي لعرض الفورم من أجل المفتاحF1، واستدعاء الطريقة ShowHelpمن معالج حدث الفورم KeyDown.

```
Private Sub Form1_KeyDown(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyEventArgs) _
Handles Me.KeyDown
'            استدعاء المساعدة عبر الشبكة
        If (e.KeyCode = Keys.F1) Then
            Windows.Forms.Help.ShowHelp(Me, "Simple.chm", HelpNavigator.Topic, "moreinfo.htm")
        End If
    End Sub
```

وكما تعلم يجب عليك وضع خاصية الفورم KeyPreview إلى صواب Trueلإطلاق الحدث KeyDown على مستوى الفورم.وإلا،فجميع المفاتيح تذهب إلى الأداة الفعالة وتتجاهل الأحداث على مستوى الفورم.

توفر الطريقة ShowHelpالكثير من الأدوات الأخرى لتجربة المستخدم من أجل مساعدة المستخدم عبر الشبكة وبما أنك أنت(وليست الأداة HelpProvider) تحدد متى يمكن الوصول إلى ملف المساعدة.

مشروع.Project

حالما تتمكن من الوصول إلى ملف المساعدة عبر الشبكة،فإنك تتمكن من الوصول إلى كل صفحة فيه.وهذا شيء جيد،لأن المستخدمين فضوليين.ولكن في حالة مشروع المكتبة،يمكن أن يقود الفضول إلى مواضيع لا علاقة لها بالزبائن العاديين.معظم الميزات في تطبيق المكتبة من أجل استخدام المستخدم الإداري.لحفظ الأشياء هادئة قدر الإمكان،يتضمن مشروع المكتبة ملفي مساعدة عبر الشبكة:

. *LibraryBasic.chm* ملف المساعدة الذي يستهدف المستخدم والذي يشرح فقط أجزاء البرنامج التي يمكن للمستخدم الوصول لها.

. *LibraryAdmin.chm*.ملف يستهدف الإداريين وأمناء المكتبة والذي يشرح كامل ميزات التطبيق.

وهذا المقطع يبني ملفات المساعدة عبر الشبكة هذه،و يكاملها مع تطبيق المكتبة.

بناء ملفات المساعدة .Building the Help Files

عملت على كتابة كلا الملفين من أجل ملفات المساعدة عبر الشبكة من أجلك.ستجدها في الدليل *Online Help*،ويتضمن التفرعين من أجل الملف والملف .يبين الشكل التالي قائمة بالملفات الموجودة في الدليل.

ChangeUser	LibraryBasic	MainForm_Basic	PatronAccess
CheckLookup_Out	LibraryBasic Indx	MainForm_Help	PatronPassword
Copyright	LibraryBasic	MainForm_Library	PatronRecord
ItemLookup	LocateDatabase	MainForm_Out	Welcome
LibraryBasic	LocateLicense	MainForm_Patron	

ملف المساعدة الخاص بالزبون.

Author	CheckLookup_Out	LibraryAdmin	NamedItem
AuthorAddLocate	CodeAuthorType	LibraryAdmin Indx	Patron
AuthorLimit	CodeCopyStatus	LibraryAdmin	PatronAccess
BarcodeLabel	CodeLocation	ListEditRecords	PatronLimit
BarcodePage	CodeMediaType	LocateDatabase	PatronPassword
BarcodePrint	CodePatronGroup	LocateLicense	PatronPayment
BarcodeToolBarcode	CodeSeries	MainForm_Admin	PatronRecord
BarcodeToolDelete	Copyright	MainForm_Basic	Publisher
BarcodeToolDown	GroupName	MainForm_Daily	PublisherAddLocate
BarcodeToolLine	Holiday	MainForm_Help	PublisherLimit
BarcodeToolNumber	ItemAuthorEdit	MainForm_In	ReportSelect
BarcodeToolRect	ItemCopy	MainForm_Library	SeriesAddLocate
BarcodeToolText	ItemLimit	MainForm_Out	SubjectAdd
BarcodeToolUp	ItemLookup	MainForm_Patron	SystemValue
ChangeUser	KeywordAdd	MainForm_Print	UserName
CheckLookup_In	LibraryAdmin	Maintenance	Welcome

ملف المساعدة الخاص بالمستخدمين الإداريين وأمناء المكتبة.

معظم ملفات HTMLلديها وصلة واحد – إلى – واحد مع نماذج مختلفة في التطبيق.على سبيل المثال،يحتوي الملف *ItemLookup.htm* محتوى المساعدة عبر الشبكة من أجل الفورم *ItemLookup.vb* في التطبيق.وتظهر هذه الصفحة في كل من ملفي إصداري الإداريين والزبائن .عندما يضغط المستخدم على المفتاح F1من فورم البحث عن بند،يحاول التطبيق إظهار صفحة المساعدة عبر الشبكة*ItemLookup.htm*. إذا كان المستخدم مستخدم قياسي،فإنه يصل إلى الصفحة في الملف *LibraryBasic.chm* ، يصل المستخدمين الإداريين إلى نفس اسم الصفحة،ولكن من الملف*LibraryAdmin.chm* .

يحتوي كل مجلد مصدر مساعدة ملفات *.hhp*، *.hhc* و *.hhk*.،تعرف المشروع،والمحتويات،وتفاصيل الفهرس،على الترتيب.تتضمن نسخة الإداري أيضاً عدة ملفات صور بتنسيق جيف GIF.
عملت على ترجمة كل ملف ووضعت نسخة من ملف ***.chm*** في هذا الدليل .

إضافة دعم المساعدة إلى التطبيق.Adding Help Support to the Application

لحفظ الأشياء بسيطة ومركزية نوعاً ما،سنعمل على توظيف الطريقة ShowHelp التي شرحناها سابقاً لعرض المساعدة عبر الشبكة من أجل كل فورم في التطبيق. عملت جميع التغيرات الضرورية وأضفت الكود من أجلك راجع المشروع.
وسأعمل على تقديم كل فورم فيما يلي.
الفورم توفر سابقاً طريقة للمدير من أجل تعين موضع كل ملف مساعدة عبر الشبكة.فهي تحدث إعدادين من خلال الكائن My.Settings.

```
My.Settings.HelpFile = Trim(RecordBasicHelp.Text)
My.Settings.HelpFileAdmin = Trim(RecordAdminHelp.Text)
```

وهذين الإعداديين تم تخزينهما في متغيرين في متغيرين شاملين.

```
MainHelpFile = RecordBasicHelp.Text
MainAdminHelpFile = RecordAdminHelp.Text
```

تلك الوسيلة الوحيدة التي نحتاجها لاستدعاء ShowHelp من كل فورم والوصول إلى واحد من الملفين متى ما ضغط المستخدم على المفتاح F1.
ولكن ماذا بخصوص لو لم يستخدم المدير الفورم *Maintenance.vb* لتركيب مواقع ملفات المساعدة؟بما أن ملفات المساعدة ومن المحتمل سيتم تنزيلها في نفس المجلد لملف البرنامج *Library.exe*، سنجدها هناك بشكل تلقائي.تعمل الطريقة InitializeSystem في الوحدة ا لبرمجية *General.vb* مسبقاً على وضع متغيرين عامين إلى قيم مخزنة في الإعدادات.

```
'تحديد ملف المساعدة عبر الشبكة هنا
        MainHelpFile = My.Settings.HelpFile & ""
        MainAdminHelpFile = My.Settings.HelpFileAdmin & ""
```

فقط في حال كانت هذه الإعدادات غير موجودة،لنضيف بعض الكود بعد هذين السطرين لتوفير الوصول الافتراضي إلى الملفات.

```
If (MainHelpFile = "") Then MainHelpFile = My.Computer.FileSystem.CombinePath( _
            My.Application.Info.DirectoryPath, "LibraryBasic.chm")
        If (MainAdminHelpFile = "") Then MainAdminHelpFile = My.Computer.FileSystem.CombinePath( _
            My.Application.Info.DirectoryPath, "LibraryAdmin.chm")
```

بما أننا بحاجة لتكييف حالة التطبيق بالنسبة للمستخدم الحالي بشكل مستمر (فيما إذا كان المستخدم زبون أو مدير)،فروتين مركزي والذي يعرض المساعدة من الملف الصحيح سيبدو أفضل.إليك الكود من أجل المساعدة عبر الشبكة،وهي طريقة جديدة في الوحدة البرمجية *General.vb* .

```
Public Sub OnlineHelp(ByVal whichForm As System.Windows.Forms.Form, ByVal contextName As String)
 ' إظهار المساعدة عبر الشبكة.التمييز بين استخدام المساعدة
 ' الأساسية والإدارية بالنسبة للمساعدة عبر الشبكة
        Dim fileToUse As String
 ' أي ملف سيتم استخدامه
        If (LoggedInUserID = -1) Then
            fileToUse = MainHelpFile
        Else
            fileToUse = MainAdminHelpFile
        End If
        If (fileToUse = "") Then
            MsgBox("لم يتم تركيب المساعدة عبر الشبكة بشكل مناسب", MsgBoxStyle.OkOnly Or
MsgBoxStyle.Exclamation, _
            ProgramTitle)
            Return
        End If
 ' إظهار المساعدة عبر الشبكة
        Try
            Help.ShowHelp(whichForm, fileToUse, _
            HelpNavigator.Topic, contextName)
        Catch
            MsgBox("حدث خطأ أثناء محاولة الوصول إلى " & _
            "ملف المساعدة عبر الشبكة", MsgBoxStyle.OkOnly Or _
            MsgBoxStyle.Exclamation, ProgramTitle)
        End Try
    End Sub
```

المهمة الأكبر في هذا الفصل تتضمن الذهاب إلى كل فورم في المشروع وعمل كل من هذين التغيرين:
.وضع خاصية الفورم KeyPreview إلى صواب True.
.إضافة استدعاء إلى OnlineHelp من معالج حدث الفورم KeyDown.
إليك الكود المضاف إلى الفورم *ChangeUser.vb*:

```
Private Sub ChangeUser_KeyDown(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyEventArgs) _
Handles Me.KeyDown
 ' إفا لإظهار المساعدة عبر الشبكة
        If (e.KeyCode = Keys.F1) Then _
        OnlineHelp(Me, "ChangeUser.htm")
    End Sub
```

العديد من النماذج تعالج متطلبات المساعدة عبر الشبكة بشكل مختلف قليلاً عن بعضها الآخر.

فالفورم *About.vb*لاتحتوي صفحة مساعدة عبر الشبكة خاصة بها،بالمقابل،فهي تعرض*Welcome.htm* .
والفورم *Splash.vb*لاتحتوي أي مساعدة بما أن المستخدم ليس من المفروض التفاعل معها.
الفورم *ReportBuiltInViewer.vb* تلك الفورم التي تظهر كل من التقارير الخمس الجاهزة،تعرض المساعدة من أجل الفورم المناسبة بواسطة *ReportSelect.htm* .
للفورم *CheckLookup*.vb صفحتي مساعدة عبر الشبكة:واحدة من أجل البنود المستعارة وأخرى من أجل البنود المعادة.
ومعالج حدثها KeyDownيختار الصفحة الصحيحة بالاعتماد على النمط الحالي من الفورم:

```
If (e.KeyCode = Keys.F1) Then
If (CheckInMode = True) Then
OnlineHelp(Me, "CheckLookup_In.htm")
Else
End If
End If
```

الفورم الرئيسية*Main.vb* أكثر تنوعاً،فهي تختار من بين تسع صفحات مساعدة عبر الشبكة عندما تكون في النمط الإداري.كل لوحة على الفورم الرئيسية مشابهة لكل فورم منفصلة،لذلك عملت على إضافة صفحة المساعدة لكل لوحة.يعمل معالج حدث الفورم على إظهار الصفحة الصحيحة بالاعتماد على اللوحة المعروضة الحالية.

```
If (PanelLibraryItem.Visible = True) Then
                    OnlineHelp(Me, "MainForm_Library.htm")
                ElseIf (PanelPatronRecord.Visible = True) Then
                    OnlineHelp(Me, "MainForm_Patron.htm")
                ElseIf (PanelHelp.Visible = True) Then
                    OnlineHelp(Me, "MainForm_Help.htm")
                ElseIf (PanelCheckOut.Visible = True) Then
                    OnlineHelp(Me, "MainForm_Out.htm")
                ElseIf (PanelCheckIn.Visible = True) Then
                    OnlineHelp(Me, "MainForm_In.htm")
                ElseIf (PanelAdmin.Visible = True) Then
                    OnlineHelp(Me, "MainForm_Admin.htm")
                ElseIf (PanelProcess.Visible = True) Then
                    OnlineHelp(Me, "MainForm_Daily.htm")
                ElseIf (PanelReports.Visible = True) Then
                    OnlineHelp(Me, "MainForm_Print.htm")
                Else
                    OnlineHelp(Me, "MainForm_Basic.htm")
                End If
```

لوحة المساعدة(أو التعليماتHelp) على الفورم الرئيسية تتضمن أزرار مصممة للقفز إلى جدول المحتويات contents وفهرس indexملف المساعدة عبر الشبكة الحالي.عملت على إضافة معالجات الحدث من أجل هذه الأزرار.
الكود من أجل كل من MainForm.ActHelpContents_Click و MainForm.ActHelpIndex_Click مشابه تماماً للكود في الروتين OnlineHelpالشامل،ما عدا الاستدعاء النهائي للطريقة ShowHelp.

```
Private Sub ActHelpContents_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles 
ActHelpContents.Click
' إظهار جدول المحتويات للمساعدة عبر الشبكة
        Dim fileToUse As String

' أيا من الملفات سيتم استخدامه
        If (LoggedInUserID = -1) Then
            fileToUse = MainHelpFile
        Else
            fileToUse = MainAdminHelpFile
        End If
        If (fileToUse = "") Then
            MsgBox("Online help is not properly configured.", _
                MsgBoxStyle.OkOnly Or MsgBoxStyle.Exclamation, ProgramTitle)
            Return
        End If

' إظهار المساعدة عبر الشبكة
        Try
            Help.ShowHelp(Me, fileToUse, HelpNavigator.TableOfContents)
        Catch
            MsgBox("An error occurred while trying to access " & _
                "the online help file.", MsgBoxStyle.OkOnly Or _
                MsgBoxStyle.Exclamation, ProgramTitle)
        End Try
    End Sub
Private Sub ActHelpIndex_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles 
ActHelpIndex.Click
' إظهار فهرس المساعدة عبر الشبكة
        Dim fileToUse As String
' أي من الملفات سيتم استخدامه
```

```
        If (LoggedInUserID = -1) Then
            fileToUse = MainHelpFile
        Else
            fileToUse = MainAdminHelpFile
        End If
        If (fileToUse = "") Then
            MsgBox("Online help is not properly configured.", _
                MsgBoxStyle.OkOnly Or MsgBoxStyle.Exclamation, ProgramTitle)
            Return
        End If
        ' إظهار المساعدة عبر الشبكة
        Try
            Help.ShowHelp(Me, fileToUse, HelpNavigator.Index)
        Catch
            MsgBox("An error occurred while trying to access " & _
                "the online help file.", MsgBoxStyle.OkOnly Or _
                MsgBoxStyle.Exclamation, ProgramTitle)
        End Try
    End Sub
```

حالما تكون ملفات المساعدة (.chm) في مكانها،وحالما يتم تركيب التطبيق بشكل مناسب لإيجاد هذه الملفات على محطة العمل(أو الشبكة المحلية)،بإمكان المستخدم الوصول للمساعدة من أي فورم بالضغط على المفتاح F1.

النشر(أو التوزيع).Deployment

تتضمن الفيجوال أستوديو خيارات مختلفة تتيح لك تنصيب تطبيقاتك المترجمة ودعم الملفات على الجهاز الهدف.سنلقي نظرة على هذه الطرق في هذا الفصل،ونستخدم واحدة من الطرق لبناء برنامج"التثبيت" من أجل مشروع المكتبة.

ما الذي يتم تضمينه في التوزيع؟.What's Involved in Deployment?

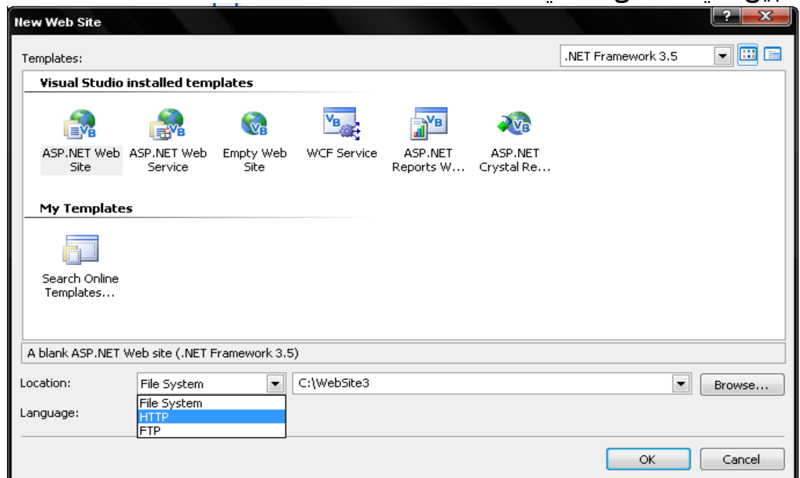
في الأيام التي سبقت ميكروسوفت ويندوز،لم يكن التوزيع صعب جداً.فالعديد من البرامج لم تكن أكثر من ملف تنفيذي لميكروسوفت دوس MS-DOS، مع واحد أو اثنين من الدعم للبيانات وملفات المساعدة.هكذا كان الأمر.حالما ما تنسخ هذه الملفات ضمن مجلد ما على جهاز العميل وتحدِّث متغير بيئة المسارPATH، سيكون الأمر قد انتهى.
تطبيقات ويندوز(وبرامج ميكروسوفت دوس الكبيرة والمعقدة)لم تكن سهلة التنصيب.فغالباً كان لديها أشياء ملف DLLمتعلقة بها-ملفات يجب أن يتم وضعها في أماكن مناسبة.وفي بعض الأحيان لاتعرف ما هو ذلك المكان المناسب،بما أنه من الممكن أن يكون بائع آخر(أو مشارك غريب)قد زود DLLبدون توثيق كافي.ومن ثم كانت ملفات التركيب(أو الإعدادconfiguration files)،التي تدعم ملفات البيانات،وتغيرات خاصة بالجهاز والمستخدم لمسجل النظامsystem registry ،الاختصارات على سطح المكتب وقائمة إبدأ،والبرامج وإعدادات إزالة التنصيب،مجموعتين من نماذج مكتبة الكونغرس(في ثلاث نسخ أصلية)،ملفات المساعدة عبر الشبكة،ملفات الترخيص واقرأني من أجل النشر على السيديات، الخطوط الخاصة والتي يمكن أن تكون مطلوبة من أجل البرنامج،وأكثر وهكذا...إلخ.
لحسن الحظ،ستشاركك الفيجوال أستوديو العبء بشكل تبادلي من أجل تسهيل التركيب عليك.
توفر لك ميزات النشر في الفيجوال أستوديو التخصص الوظيفي الأساسي الذي تحتاجه لنشر تطبيقات معتمدة على ويب والمعتمدة على تطبيقات سطح المكتب القياسية.إذا كانت احتياجات التوزيع الخاصة بك معقدة،تستطيع أيضاً شراء مشارك إضافي أداة"التثبيت والنشر setup and deployment " والتي تتضمن ميزات إضافية مثل دعم الصيغ(الإنشاءscripting).

طرق التوزيع ضمن الفيجوال أستوديو.Deployment Methods Within Visual Studio

مع الإصدار الأسبق من الفيجوال بيسك،إذا كنت تريد تنصيب برمجيات مخصصة باستخدام برنامج التثبيت،عليك إما كتابته بنفسك أو استخدام أداة مشترة.تظهر أدوات التوزيع في النهاية في الفيجوال بيسك،وخاصة "المعالج السحري للتحزيم والتوزيع Package and Deployment Wizard " السيئ السمعة.برنامج التثبيت المعلب هذا تم كتابته في الفيجوال بيسك،وتستطيع تحسينه ليلاقي احتياجات التوزيع الخاصة بك.
ولكن لم يكن سهل.وكان باقي العالم مسبقاً يتخذ منصة"منصب ويندوز" من أجل التوزيع الموحد المعايير بواسطة ملفات *.msi*.. يستخدم المعالج السحري للتحزيم والتوزيع تنسيق الملف *.cab* الأقدم. حتى من أجل مبرمج ما يستمع بالبرمجة حقاً،الحاجة لكتابة برامج تنصيب فعالة يجعل في بعض الأحيان الحياة قبيحة.
عندما أتت الفيجوال بيسك2002،أصبحت الحياة جميلة مرةً أخرى.تضمنت الفيجوال أستوديو أدوات تتيح لك استهداف تقنية منصب ويندوز، فقد كان نسخة محللة تتيح لك تحرير التطبيقات الأبسط فقط،ولكن كان على البائعين المشاركين أن يحوزوا بعض المتعة.
في هذه الأيام،تتضمن الفيجوال أستوديو العديد من طرق التوزيع،تقدمة للأنواع المختلفة من التطبيقات،الأنواع المختلفة من المستخدمين، والأنواع المختلفة من البيئات المأمونة التي من المحتمل أن يحتاج المبرمج استهدافها.أقرأ كل الطرق المتاحة لرؤية أياً منها تناسب احتياجات برنامجك.لقد اخترت طريقة توزيع مشروع المكتبة،ولن أبوح بها حتى منتصف هذا الفصل.

توزيع آسبي دوت نت المباشر.Direct ASP.NET Deployment

من الواضح أن تطبيقات آسبي دوت نت مختلفة عن تطبيقات سطح المكتب.واحد من الاختلافات الكبيرة هو من أجل المستخدم النهائي، تطبيقات آسبي دوت نت ليس لديها حقاً أي توزيع.عليك فقط الاستعراض إلى موقع الويب المناسب وتستخدم التطبيق.ولكن مايزال التوزيع ضروري لخادم ويب المستضيف.إذا كان قد تم تنصيب امتدادات فورنت بيج FrontPageلخادم ويب الخاص بك، تستطيع تنصيب تطبيق آسبي دوت نت المترجم من بيئة التطوير الخاصة بك بكل راحة وأمان.لقد علقت عليه في الفصل23،ولكن الفيجوال أستوديو تجلب لك خيار وضع تطبيق ويب على موقع ويب موجود حقاً عندما تحاول للمرة الأولى إنشاء تطبيق آسبي دوت نت.في نموذج موقع ويب جديد،تستطيع اختيار HTTP URL كموقع تطوير،كما هو مبين في الشكل التالي.



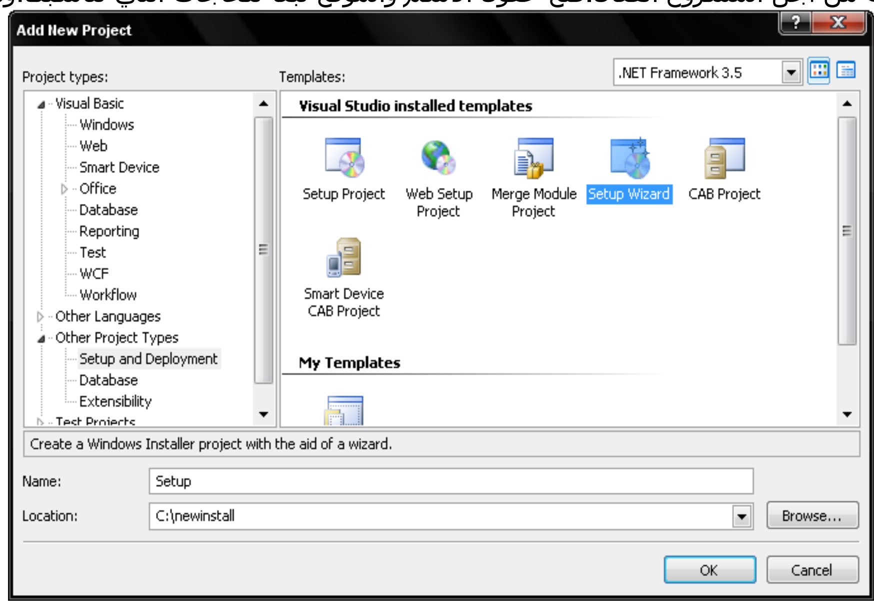
بما أنك ستطور موقع ويب الخاص بك بشكل تفاعلي،من المحتمل أنك لاتريد استخدام هذه الطريقة على خادم الإنتاج،بالمقابل،تستطيع التطوير بشكل محلي في دليل أو على خادم تطوير ويب،ومن ثم فيما بعد تنشر *publish*الموقع إلى خادم الإنتاج.وهذا سهل بقدر إعداد موقع HTTP(بروتوكول نقل النص الفائق Hypertext Transport (or Transfer) Protocol)من البداية.مع موقع الويب في الفيجوال أستوديو،اختر القائمة بناء Build >> نشر موقع ويبPublish Web Site. وعين عنوان صفحة الويب(URL)uniform (or universal) resource locator.وليس هناك حاجة لبرنامج تثبيت منفصل.إن آسبي دوت نت حريص فيما يخص كيفية معالجته للملفات في تطبيقك.ولن ينشر كودك المصدري. سينسخ ملف *web.config* إلى الخادم(وهو ملف مطلوب)،والذي يمكن أن يحتوي على نص اتصال قاعدة بياناتك.ولكن خادم ويب لأسبي دوت نت المركب كما ينبغي سيحفظ هذا الملف بعيداً عن عيون المتطفلين.

توزيع ×نسخة.XCopy Deployment

تحتوي مجمعات الدوت نت المترجمة كشف يصف المجمع وحاجياته.وهذا يعني أن باستطاعتك نسخ المجمع إلى نظام أخر قد تم تنصيب إصدار إطار العمل المناسب،وطالما أن الملفات الأخرى التي يحتاجها المجمع تم نسخها أيضاً،فإن البرنامج سيعمل.وهذا يدعى"التوزيع بـ×نسخةXCopy deployment "،لأنك تستطيع استخدام سطر الأمر XCopyلنقل الملفات.من المحتمل أنك تفكر،حسناً،إن مجمع EXEهو برنامج ويندوز حقيقي.بالطبع سيعمل عندما أنسخه إلى نظام جديد.حسناً وكل هذا صحيح.ولكنه لم يكن صحيح بالنسبة لتطبيقات الفيجوال بيسك الأقدم.فأدوات المستخدمة من قبل تطبيقات الفيجوال بيسك المعتمدة على المكونات يجب أن يتم تسجيلها في مسجل ويندوز قبل أن يكون بالإمكان الوصول إليها وقت التنفيذ.وتتطلب برامج الفيجوال بيسك الأقدم أيضاً أن يتم تنصيب مكتبات وقت التشغيلruntime libraries .
ويجب أن يتم تنصيب أيضاً إطار عمل الدوت نت من أجل برامج الدوت نت .ولكن وبما أن إطار العمل يتم إدارته بشكل آلي بواسطة نظام تحديث ويندوز،فهذه ليست بالمشكلة العظيمة.في معظم الحالات ما أود قوله في هذه الجمل،أنّ بإمكانك تنصيب تطبيق الدوت نت على جهاز ما بنسخ البرنامج فقط،ومن الممكن عدة ملفات دعم،إلى دليل ما.ولا أقول أنها الطريقة التي عليك استخدامها لتنصيب البرامج.عملياً،سأصدم إذا ما اكتشفت أي صديق مبرمج يستخدم هذه الطريقة في بيئة العمل الفعلية.ولكن الدوت نت جعلت خيار التوزيع هذا متاح لك إذا كنت لاتريد أن تكون صديقي بعد الآن.فإذا كنت تريد استخدام النشر ×نسخة،من المحتمل أنه ليس لديك أي مشكلة مع الأمن،أو التحديدات الإدارية والتي من الممكن أن تفرض على الجهاز.فالفرص هي،إذا ما نصبت برمجيات باستخدام الأمر ×نسخة،أو بسحب وإسقاط الملفات،من المحتمل ولأنك صديق مع مالك الجهاز.

توزيع منصب ويندوز.Windows Installer Deployment

إن منصب ويندوز هو نظام تنصيب مكتبي موفر من قبل ميكروسوفت.وهو يخدم كنظام قاعدي لحزم التنصيب الناتجة عن الفيجوال أستوديو القياسي،ويوفر أيضاً الدعم أساسي underpinningsلمعظم أدوات التنصيب الأجنبية.قبل منصب ويندوز،كل بائع حزمة تنصيب يعمل شيء إلى حدٍ ما يراه مناسب.ولكن هذا عنى أن منتجات التنصيب في بعض الأحيان تضرب clobberedبعضها بعضاً،بما أن حزمة البرمجيات software package ليس من الضروري أن تبحث عن ملفات تم تنصيبها بواسطة أداة أخرى.وإصلاح مثل هذا الضرر كان صعباً بالنسبة للمستخدم،فمن عادة لايريد حتى أن يعرف أي الملفات التي تم تنصيبها أو تحديثها.
سعت ميكروسوفت لتغير ذلك مع منصب ويندوز. واحدة من الميزات الرئيسية للنظام وهي قاعدة بيانات الملفات المنصبة والمحدثة.ويدعم منصب ويندوز أيضاً إمكانية كاملة لإزالة التنصيب/الترميم والمسار الراجع(التراجعrollback (الارتداد)) لذلك فيمكن التراجع عن أي فشل،إصلاح (أو إعادة restoring)النظام إلى حالته السابقة.وميزات أخرى تتضمن دعم تصحيح البرامج من الأخطاءpatching، إعادة التشغيل rebooting،إصلاح أو" معالجةheal " التنصيب السابق ولكن البرنامج المتضرر،والتنصيب حسب الحاجة(أو حسب الطلبinstall-on-demand) والذي يحتفظ بميزات أو بكامل التطبيقات على وسيلة التنصيب حتى يحاول المستخدم استخدام تلك الميزة.
منصب ويندوز النسخة 4.*x*هي الإصدار الأخير من أجل ويندوز فيستا وأنظمة ويندوز الموازية(ما يزال بإمكانك الحصول على الإصدار 3.*x*من أجل ويندوز XP أو 2.*x*من أجل بعض أنظمة ويندوز الأخرى مثل ويندوز98).
إن جوهر نظام منصب ويندوز هو ملف " MSI "(مع ملف الامتداد *.msi*)،الملف الوحيد الذي يحتوي جميع الملفات والتعليمات(التوجيهات) الضرورية لتنصيب، تحديث،وإزالة تنصيب منتج برمجي.تستطيع الفيجوال أستوديو إنشاء مشاريع تثبيت بالاعتماد على معيار MSI،على الرغم من أنك لاتستطيع استخدام بعض الميزات الأكثر تقدماً لمنصب ويندوز من خلال الفيجوال أستوديو.إذا ما كانت احتياجاتك بسيطة ومعظم برمجيات العمل المكتوبة في الفيجوال بيسك لديها احتياجات تنصيب بسيطة- فما تزال الفيجوال أستوديو تفي بالغرض.
بناء مشروع تثبيت سهل تماماً بقدر سهولة إنشاء مشروع تطوير فيجوال أستوديو قياسي.ولكن أولاً،احتاج شيء ما لتثبيته.من أجل المناقشة في هذا الفصل،عملت على إنشاء تطبيق سطح مكتب.بكل بساطة عملت على إنشاء مشروع *WindowsApplication1*جديد مع نموذجه الافتراضيForm1، وحفظته إلى المجلد*C:\temp*. كل ما يعمله عندما يتم تشغيله هو عرض نموذجه الافتراضيForm1 .
لإنشاء ملف التنصيب من أجل مشروع الفيجوال بيسك،افتح ذلك المشروع في الفيجوال أستوديو واستخدم القائمة ملف File >>إضافةAdd >>مشروع جديدNew Project، لإضافة مشروع تثبيت إلى كامل الحل بحيث يحتوي على مشروعك الأصلي.يبين الشكل التالي حوار إضافة مشروع جديد.اختر نوع المشروعproject types: تثبيت ونشرSetup and Deployment، ومن ثم القالب "المعالج السحر للتثبيت Setup Wizard" لإنشاء برنامج التثبيت من أجل المشروع الفعال.ضع حقول الاسم والموقع تبعاً للحاجات التي تناسبك،ومن ثم انقر موافقOK .



يظهر معالج التنصيب السحري،يقودك خلال خمسة خطوات:

الخطوة1.

الخطوة الأولى للمعالج تقول فقط"مرحباً"لذلك انقر الزر التالي واذهب إلى العمل الحقيقي.

الخطوة2.

تطلب الخطوة 2منك نوع مشروع التثبيت الذي سينتج.شخصياً،اعتقد أنه من الممكن معرفته من محتوى المشاريع المحملة سابقاً،ولكن إذا فعل المعالج السحري كل شيء،لما سيحتاج العالم مبرمجين مثلنا؟توجد أربع خيارات مبينة في الشكل التالي.

Do you want to create a setup program to install an application?

- Create a setup for a Windows application
- Create a setup for a web application

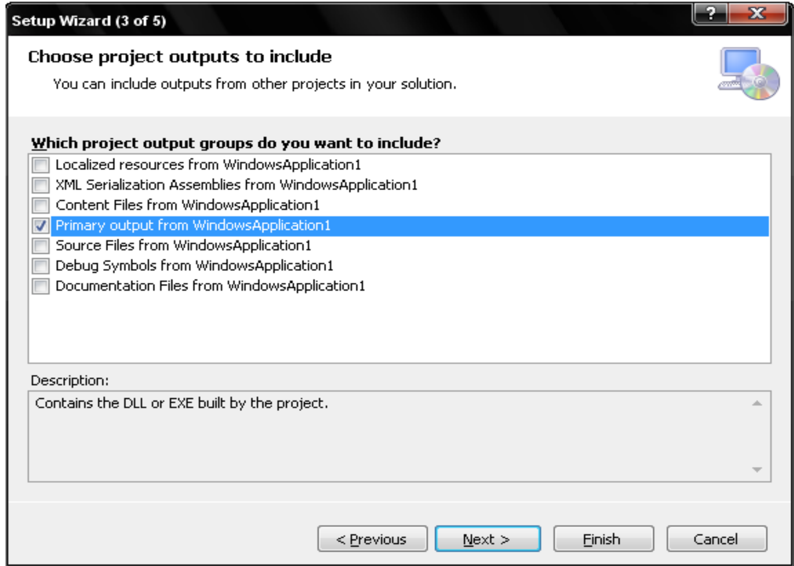
Do you want to create a redistributable package?

- Create a merge module for Windows Installer
- Create a downloadable CAB file

الخيارين الأوليين في الأعلى يعملان على إنشاء ملفات تنصيب كاملة إما لتطبيقات سطح المكتب أو للويب.(التثبيت المعتمد على ويب سيتم تسليمه إلى مدير موقع ويب للتنصيب على الخادم).تتيح لك وحدات الدمج *Merge modules* إنشاء حصة من تنصيب ما يمكن أن يتم دمجها فيما بعد في ملف MSIكامل.وهذا خيار جيد إذا كنت تصمم مكتبة library سيتم استخدامها من أجل تطبيقات متعددة،ولكنه غير مفيد بحد ذاته(أو لوحده).الخيار الأخير ملف CABيعمل على إنشاء أرشيف archiveمن الملفات يمكن أن يتم تنصيبها باستخدام تقنية نشر ملف أقدم بعض الشيء. وهو أيضاً نظام النشر المستخدم لأجهزة الكمبيوتر الصغيرة.بما أنني استهدف تطبيق سطح المكتب،سأختار الخيار الأول"إنشاء تثبيت من أجل تطبيق ويندوز" وانقر التالي Next .

الخطوة 3.

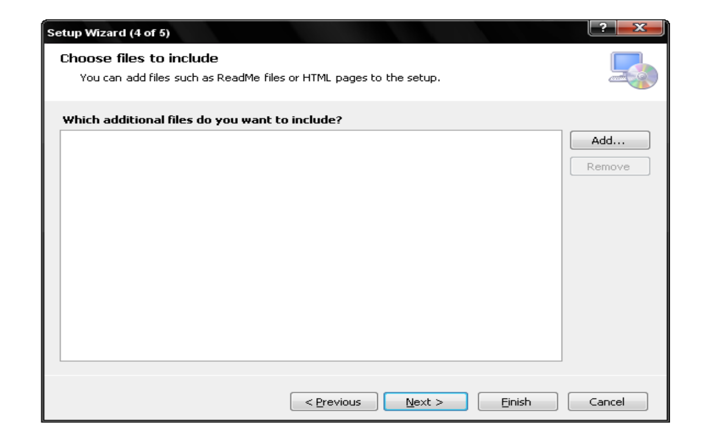
على الرغم من أنك تستطيع إنشاء برنامج تثبيت يعمل ببساطة على تنصيب ملفات شتى تقتات scavenged من قرصك الصلب،فأنت عادة تبني مشروع تثبيت بالاعتماد على ملفات أو مخرجات مترجمة لمشاريع أخرى.تطلب الخطوة الثالثة للمعالج السحري منك تضمين العناصر من مشاريع أخرى موجودة في حل الفيجوال أستوديو الفعال.لقد اخترت تضمين ملف EXE المترجم من مشروع سطح مكتبي،كما هو مبين في الشكل التالي.بشكل عام لا أعمل على تضمين كودي المصدري في مشروع التثبيت،لذلك سأترك تلك العناصر غير مختارة.ولكن البند ملفات المحتوى Content Files يمكن أن تكون مفيدة.إذا كان مشروعي يحتوي على ملف مساعدة عبر الشبكة(مع امتداد الملف *.chm*)،بإمكاني إضافته كملف محتوى قياسي إلى المشروع الرئيسي بواسطة القائمة مشروع Project >>إضافة بند موجود Add Existing Item .



سيتم تصنيف (أو تبويب)ذلك الملف كمحتوى،ويمكن نقله إلى مشروع التثبيت من خلال اختيار ملفات المحتوى Content Files. ولكن توجد طرق أخرى لتضمين مساعدة عبر الشبكة في التنصيب،والتي سنراها في الخطوة التالية.الآن،سنبقى على الاختيار Primary output،وانقر التالي Next.

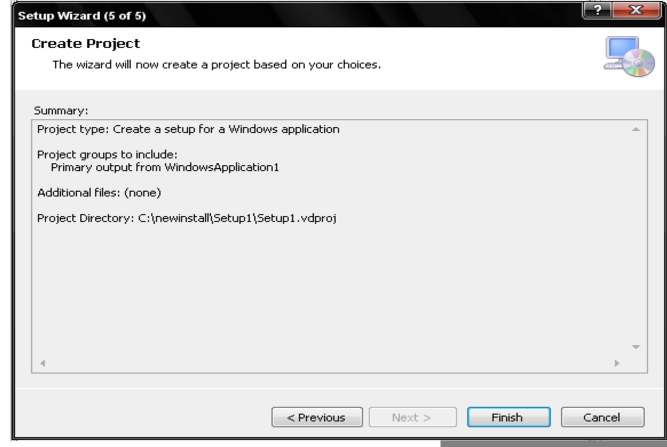
الخطوة4.

في هذه الخطوة،تستطيع إضافة أي ملفات غير خاصة بمشروع معين أخرى والتي تريدها إلى مشروع التثبيت(شاهد الشكل التالي).ملفات اقرأني Readme،محتوى المساعدة عبر الشبكة،اتفاقيات الترخيص license agreements ، صورة لأطفالك،وإلى حدٍ ما أي شيء أخر يمكن أن يتم تضمينه هنا.لن أعمل على إضافة أي شيء أكثر.أنقر التالي Next .



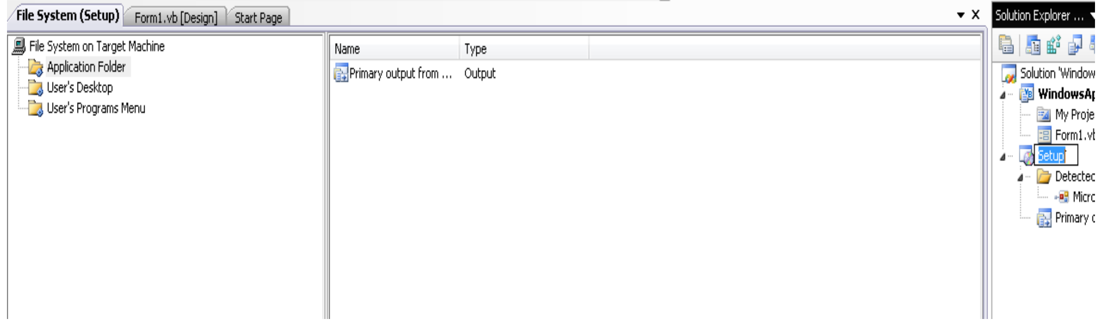
الخطوة 5.

تعرض الخطوة الأخيرة ملخص للاختيارات التي عملته(شاهد الشكل التالي).حسناً،المعالج السحري سهل جداً.فيقع على عاتقنا العمل في ثلاث خطوات من الخطوات الخمس.انقر إنهاء لإكمال المعالج السحري.



بعد المعالج السحري.After the wizard

حالما يكتمل المعالج السحري،تظهر الواجهة الرئيسية لتصميم مشروع التثبيت للفيجوال أستوديو في نافذة التطوير.يبين الشكل التالي الفيجوال أستوديو وهي تعرض مشروع إنتاج التثبيت الأحدث لـ *WindowsApplication1*،يظهر أيضاً مشروع أخر في لوحة مستكشف الحلول.



النافذة الرئيسية في الشكل السابق واحدة من عدة محررات editors تتيح لك تخصيص مشروع التثبيت.تستطيع الوصول إلى كل محرر من خلال القائمة عرض View >> المحررEditor. أو باستخدام أزرار شريط الأدوات في لوحة toolbar مستكشف الحلولSolution Explorer panel .

محرر نظام الملفات.File System Editor

إنه ذلك المحرر الذي يظهر في الشكل السابق.وهو يجلب مجلد قياسي/عرض بند لحصة نظام ملفات filesystem للنظام الهدف.من خلال هذه الطبقة الهرمية،تضع الملفات(مخرجات EXE من مشروعك الرئيسي،ملفات المساعدةhelp files، ملفات التركيبconfiguration ، المختصرات shortcutsلأي من هذه الملفات،إلخ)في مجلدات خاصة(مجلد التطبيقApplication folder، سطح المكتبDesktop، ملفات البرنامج 64-بت أو 32-بت32- or 64-bit Program Files، الخطوطFonts، مجلد قائمة إبدأStart Menu folder، وأخريات).إذا كنت لا ترى المجلد الذي تريده في نظام الملفات على لوحة الجهاز المستهدف،استخدم القائمة إجراء Action >>إضافة مجلد خاص Add Special Folder لتضمينه في القائمة.بالإضافة إلى المجلدات الخاصة القياسيةstandard special folders، تتضمن القائمة إضافة مجلد خاص Add Special Folder خيار مجلد مخصص Custom Folderتتيح لك إنشاء مجلد معين specific folder في أي مكان على النظام الهدف.

محرر التسجيل.Registry Editor

يعرض هذا المحرر تسلسل هرمي مبتور لخلايا المسجل.إضافة أية مفاتيح أو قيم هنا سيتم إنشاءها في مسجل المستخدم user's registry خلال التنصيب.

محرر أنواع الملفات.File Types Editor

يتيح لك هذا المحرر تعريف مرفقات بين امتداد ملف ما(مثل *.txt*)وبرامج معينة أو إجراءات.أي إجراء مخصص،مثل فتح أو طباعة،يمكن أن يتم ربطه لأي نص أمر تريده،ومن ضمنه الأوامر التي تستهدف المجمع الرئيسي الذي سيتم تنصيبه.

محرر واجهة المستخدم.User Interface Editor

يتضمن مشروع التنصيب الافتراضي عدة نماذج تطلب أشياء مثل موقع التنصيب والمعلومات حيث يجب أن يحدث التنصيب.تستطيع إدراج صناديق حوار إضافية في سياق التنصيب.ولكن احذر:لن تكون قادر على إضافة نماذج تمكين الفيجوال بيسك كاملاً.بالمقابل،ستختار عدة حوارات مسبقة التعريف (مثل حوار اتفاقية الترخيص،أو حوار أزرار التبديل الأربع Radio Buttons)،ووضع خاصيات الحوار لتركيب نص العرض لكل حقل حوار أو طلب. كل حقل مدخلة مستخدم/أداة تتضمن قيمة بالاسم تستخدمها في المحررات الأخرى لتحديد إجراء تنصيب معين.على سبيل المثال،تستطيع مراقبة قيمة صندوق اختبار checkbox يطلب من المستخدم،وإذا لم يختاره المستخدم،فتستطيع كبح التنصيب بالنسبة لحقول معينة يتم إرفاقها مع صندوق الاختبارcheckbox .

محرر الإجراءات المخصصة.Custom Actions Editor

إذا كنت تريد مستوى نهائي لأداة،تستطيع إضافة إجراء مخصص،استدعاء لبرنامج خارجي أو صيغة،يتم تشغيله عند نقطة معينة في عملية التنصيب (أو إزالة التنصيب)

محرر شروط الإطلاق.Launch Conditions Editor

إذا كان يجب على الجهاز الهدف أن يكون في حالة معينة قبل أن تتمكن من تنصيب المشروع بشكل ناجح،فيتيح لك هذا المحرر تعريف شروط التحديد.بشكل افتراضي،يضيف المنصب إطار عمل الدوت نت كشرط تنصيب،فيجب أن يتم تنصيب إطار عمل الدوت نت قبل أن يتم تنصيب المشروع.تستطيع البحث عن ملفات معينة أو مفاتيح تسجيل والتي يجب أن توجد قبل أن يبدأ التنصيب.على سبيل المثال،من المحتمل أنك تريد التأكيد على أن مشغلات قاعدة البيانات الهدف target database drivers يجب أن تكون على النظام قبل تنصيب تطبيق معتمد على قاعدة بيانات.

إنتاج ملف Generating the MSI file.MSI

حالما تعمل على إعداد مشروعك من خلال المحررات المتنوعة،فتعمل على إخراج ملف MSI النهائي ببناء الحل بواسطة القائمة بناء Build >> بناء الحلBuild Solution . يظهر ملف MSI في الموضع المعين في خاصيات مشروع التثبيت(القائمة مشروع Project >> الخاصيات Properties)وهذا الملف يحتوي جميع الاختصارات والمحتوى المطلوب لتنصيب التطبيق بالكامل على الجهاز الهدف.

التوزيع بنقرة واحدة.ClickOnce Deployment

تتضمن الفيجوال أستوديو2008 طريقة توزيع تدعى نقرة واحدة*ClickOnce* .فقد تم تصميمها لتوفير سهولة في توزيع التنصيب النهائي من أجل تطبيقات سطح المكتب(نماذج ويندوز).وما يزال يحتوي معالج سحري،ولكن للتنصيبات القاعدية،هذا كل ما هنالك.حالما يتم توزيع تطبيقك من خلالClickOnce ،يستطيع المستخدم تنصيبه مباشرةً من موقع ويب أو موضع تخزين أخر.يبدو هذا مثل تنصيب MSIالقياسي،ولكن مختلف في عدة طرق:

▪التوزيعات ClickOnceيمكن أن يتم تنصيبها حتى ولو كان المستخدم الحالي ليس لديه امتيازات إدارية محلية.العديد من البرمجيات تنصب ملفات مفتاحيه تؤثر في مجلدات ويندوز *Windows* وويندوز/نظام32:*Windows\System32* ،أو في مجلدات أخرى هامة ولكنها مجلدات مقيدة(أو محصورة).إذا كنت مطور،من المحتمل أنك لن تجرب هذه المشكلة،لأنك المدير على جهازك الخاص.ولكن في تنظيمات دائرة تقنية معلومات المدارة مع العديد من المستخدمين،توجد منفعة لتخفيض مستوى الامتياز للمستخدمين المستقلين.تأثير واحد سلبي لهذا وهو أنه يجب أن يحضر المدير لتنصيب أي برنامج.ولكن هذه ليست هي الحالة مع ClickOnce.أي تطبيق موزع بـ ClickOnce يمكن أن يتم تنصيبه بواسطة أي مستخدم.

▪التطبيقات الموزعة بـ ClickOnce يمكن أن تعمل على إطلاق تحديثات البرمجيات الخاصة بها بشكل آلي.إذا كان التركيب بهذه الطريقة،سيتفحص البرنامج موضع التوزيع الأصلي من أجل نسخة جديدة كل مرة يتم تشغيله.فإذا كان هناك نسخة جديدة،سيتم تنصيبه بشكل آلي بدون أن يعمل المستخدم أي شيء.

▪تطبيقات ClickOnceيتم تصميمها لسهولة التنصيب.مع تطبيق التوزيعMSI،تحتاج إلى تنزيل ملف MSIومعالجته من خلال نظام منصب ويندوز.على الرغم من أن عليك أيضاً تنزيل(تحميل) توزيعClickOnce ،فهو يحدث بشفافية أكثر أو أقل.التطبيقات المنشورة بنقرة واحدة يمكن أن يتم تركيبها بحيث تبدو كامتداد صفحة ويب:نقر وصلة،والبرنامج مباشرةً يعمل،توزيع نموذجها الرئيسي للمستخدم(يمكن أن يكون هناك بعض التأخير بما أن البرنامج تم تحميله على الانترنت).إن ذلك يبدو عظيماً.ولكن وبما أن تطبيقات تمكين ClickOnce (بشكل افترضي)تعمل في صندوقها الرملي(ساعتها الرملية) الخاصة،فهي محدودة في وصولها إلى بعض الموارد المحلية.وأيضا،لدعم جميع ميزات التحديث الآلي، عليك إضافة كود لتطبيقك يقوم بعمل التحديث الحقيقي.(توفر الخاصية My.Application.Deploymentإمكانية الوصول إلى هذه الميزات)

لتوزيع تطبيقك بواسطة ClickOnce،استخدم القائمة بناء Build >> توزيع Publishفي الفيجوال أستوديو.بعد أن يسألك بعض الأسئلة الأساسية حول من أين سيحصل المستخدم على ملف التوزيع(من موقع ويبweb site ،مجلد شبكةnetwork folder ،أو سيدي/ديفيدي CD/DVD)،تعمل الفيجوال أستوديو على إنتاج ملف التنصيب وتجعله متاح مباشرةً للاستخدام.بالطبع هذه الطريقة تمنحك فقط خيارات التنصيب القاعدية.فهي تجعل ملف EXEأو DLLالرئيسي (و توابعه)لمشروعك متاحة للتنصيب على الجهاز الهدف،ولكن هذا تقريباً كل شيء.إذا كنت تريد تحكم أكثر على عملية التوزيع والمكونات التي سيتم تضمينها،استخدم تبويب التوزيع Publishلخاصيات مشروعك،كما هو مبين في الشكل التالي.

Compile
Debug
References
Resources
Services
Settings
Signing
My Extensions
Security
Publish

Publish Location
Publishing Folder Location (web site, ftp server, or file path):
http://localhost/WindowsApplication1/
Installation Folder URL (if different than above):
Install Mode and Settings
The application is available online only
The application is available offline as well (launchable from Start menu)
Application Files...
Prerequisites...
Updates...
Options...
Publish Version
Major: 1 Minor: 0 Build: 0 Revision: 0
Automatically increment revision with each publish
Publish Wizard... Publish Now

تتضمن هذه اللوحة الحقول التي تتيح لك وضع رقم الإصدار لكل حزمة تنصيب موزعة.إذا عدلت رقم الإصدار هذا وأعدت توزيع التطبيق،فإن كود التوزيع المخصص الذي أضفته إلى التطبيق يمكن أن يلتقط الإصدار الجديد ويستهل التحديث من موقع النشر.

مشروع.Project

لقد اخترت توزيع منصب ويندوز القياسي،لأنني أظنه سيوافق معظم احتياجات مستخدم نظام المكتبة النموذجي.فقد تم إعداد تطبيق المكتبة بقصد أن يكون ميزة مستقرة(ثابتة)،لذلك من المحتمل أن شخص ما مع معرفته بالمعلومات التقنية أو الامتيازات الإدارية سيقوم بعمل التنصيب الحقيقي.ما يزال التطبيق يحتوي العديد من الملفات،ومن ضمنها ملفي مساعدة عبر الشبكة،لذلك فالتنصيب س.نسخة سيكون عبء إضافي. لذلك تنصيب MSI القياسي هو خطة التوزيع الأفضل.

تخطيط التوزيع.Planning the Deployment

يعمل معالج التثبيت السحري على إضافة مجمع مشروعي بشكل آلي إلى ملف MSI،ولكنني متأكد أن ملفات أخرى ضرورية لتوزيع مشروع المكتبة بشكل مناسب.فنظرة سريعة إلى الفصول السابقة يوحي بمتطلبات القائمة التالية من الملفات التالية:

The .NET Framework 3.5

يجب أن يتم تنصيبه على الجهاز الهدف لتشغيل تطبيق المكتبة.سيحتاج برنامج الإعداد أن يتم تنصيبه بشكل آلي إذا لم يكن مسبقاً على الجهاز الهدف.

Library.exe

إنه المجمع الأساسي.سيكون التنصيب عديم الفائدة بدونه.

LibraryBasic.chm and LibraryAdmin.chm

وهي ملفات المساعدة عبر الشبكة،وسيتم تنصيبها في نفس مجلد التطبيق الأساسي.

The bar code font

إذا حصلت على حقوق توزيع خط القيم الشاملة،يستطيع برنامج التثبيت نسخه بشكل مباشر إلى مجلد خطوط نظام الجهاز الهدف.

LibraryLicense.lic

ملف الترخيص- تذكر أنه ملف تم إنتاجه بشكل يدوي ويحتاج أن يتم صنعه بالنسبة لكل زبون يشتري تطبيق المكتبة.ترجمته مباشرةً إلى برنامج التثبيت يبدو زيادة مفرطة،بما أنني سأعمل على إنتاج تثبيت لكل زبون.بالمقابل،سأضع الملف على وسيطة(أو سيدي)،وعلى المستخدم إيجاده عند تشغيل برنامج المكتبة.

ACME Library Resource Kit.pdf

هذا الملف الذي على مستوى المدير لن يتم تنصيبه بشكل افتراضي عل الجهاز.سيبقى بالمقابل على قرص التوزيع.

Database Creation Script.sql

إذا كنت أعمل على تطوير تطبيق للمستخدم النهائي،سأعمل على بناء نظام تثبيت منفصل من أجل حصة الخادم،مركزاً بشكل رئيسي على تنصيب قاعدة البيانات.بما أن هذا الكتاب تم تصميمه كمدخل فقط،سأعمل فقط على نسخ صيغة بناء قاعدة البيانات إلى قرص التوزيع وافترض أنه مؤهل لتمثيل تقنية المعلومات أو سيأخذ مدير قاعدة البيانات المسؤولية لتنصيب هذه الخطوة.

The Library web site

كما مع صيغة إنشاء قاعدة البيانات،فسأعمل على نسخ ملفات موقع الويب إلى قرص التوزيع،وادع المدير يستكشف الأشياء.

Readme.htm

سيتضمن السيدي ملف معلومات تماماً عند الجذر الذي سيخبر المستخدم كيفية استخدام الملفات على السيدي.لم أكتب هذا الملف حتى الآن،ولكن سأكتبه قبل أن ينتهي هذا الفصل.
سيتضمن ملف التثبيت الناتج فقط البنود الأربع الأولى في القائمة في الأعلى(الثالث يتضمن الخط)،والاثنين الأوليين يتم إضافتهم بشكل آلي من قبل معالج التثبيت السحري.إن كل ما سبق لن يكون صعباً جداً.

بناء مشروع التثبيت.Building the Setup Project

مسبقاً في هذا الفصل،عملت على إضافة مشروع جديد إلى مشروع موجود،وضممتهم إلى حل واحد.من الممكن بناء مشروع تثبيت بحيث يبدو قائم بنفسه ضمن الفيجوال أستوديو.في مثل هذه المشاريع،تحتاج إلى استعراض لإيجاد المجمع الهدف(*release\Library.exe*)لتضمينه في مخرجات التثبيت.مهما يكن،لايعمل معالج التثبيت السحري الكثير لك إذا كنت ستسلك ذلك الطريق.لذلك،من أجل مشروع المكتبة،دعنا نضيف مشروع تثبيت جديد إلى مشروع المكتبة المحمل ضمن الفيجوال أستوديو.
الخطوة الأولى للخطوات المتعددة المتوازية مع الخطوات التي عملناها سابقاً في هذا الفصل.فحالما تحمل مشروع المكتبة وتحفظه إلى مجلده الهدف،أضف مشروع تثبيت باستخدام القائمة ملف File << إضافة Add << مشروع جديدNew Project . اختر معالج التثبيت السحري

Setup Wizard كقالب،ادخل في حقل الاسم**LibrarySetup**، واستخدم مجلد مشروع المكتبة لحفظ الموقع.طبق الإعدادات التالية ضمن المعالج:

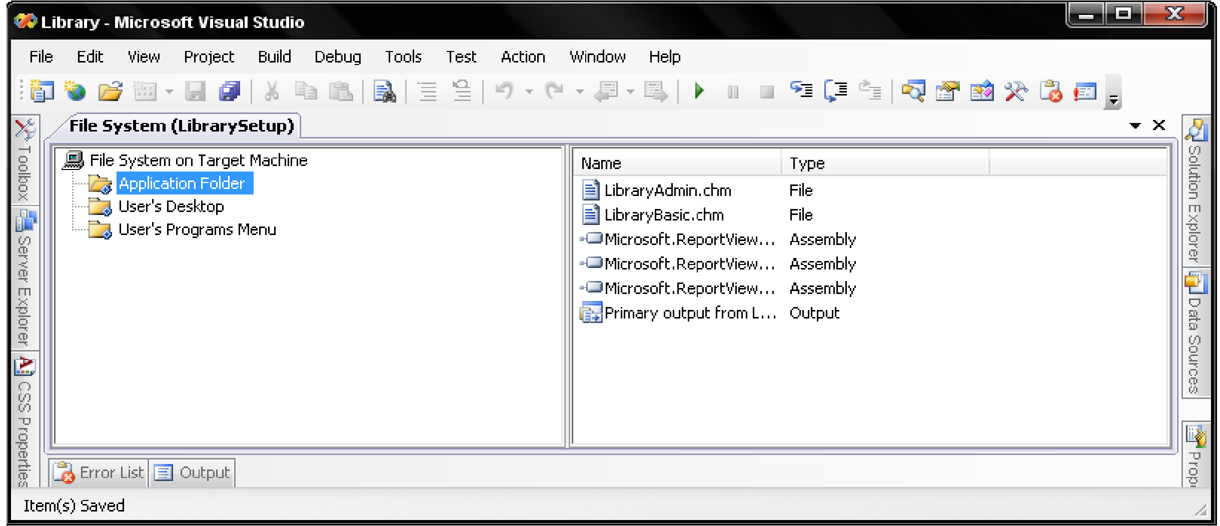
- في الخطوة الثانية،اختر"إنشاء تثبيت لتطبيق ويندوز Create a setup for a Windows application "
- في الخطوة3،اختر"مخرجات رئيسية من المكتبةPrimary output from Library "من القائمة.
- في الخطوة 4،اعمل على إيجاد الملف*LibraryBasic.chm* والملف*LibraryAdmin.chm* .في دليل تنصيب الكتاب،تستطيع إيجادها في الدليل الفرعي المسمى *Online Help*.

أكمل المعالج السحري واستخدم القائمة ملف File >>حفظ الجميعSave All. عندما تسأل عن حفظ ملف الحل(*Library.sln*)،خزنه فقط في دليل مشروع المكتبة،والذي سيكون مختار مسبقاً.

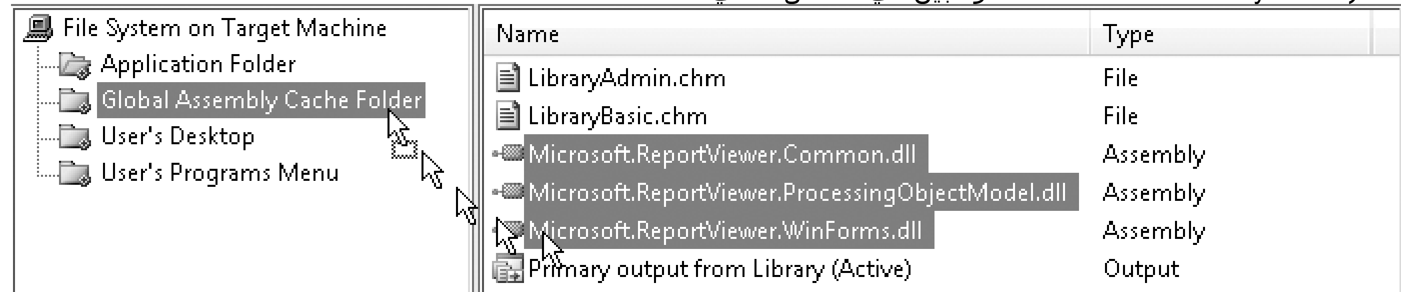
كما سبق،يفتح مشروع التثبيت إلى محرر نظام الملفthe File System Editor. قبل عمل أي تغيرات ضمن المحرر،دعنا نضع بعض خاصيات التثبيت الواسعة.انقر على تثبيت المكتبة LibrarySetupفي لوحة مستكشف الحلول، وعدل الخاصيات التالية في لوحة الخاصياتProperties :

- ضع خاصية المؤلف Authorإلى"اسمك الخاص"أو ما تشاء.
- ضع خاصية الصانع Manufacturerإلى " ACME ".
- ضع خاصية عنوان صفحة الصانع ManufacturerURLإلى " http://www.MHM.com "أو أي موقع ويب ترغب باستخدامه.
- ضع الخاصية "اسم المنتج ProductName "إلى " ACME Library ".
- ضع خاصية العنوان Titleإلى"تنصيب المكتبة".

بما أن محرر نظام الملفات File System Editor مفتوح،دعنا نعمل العديد من التغيرات هناك.عندما أضفنا المجمع *Library.exe* خلال المعالج السحري،فإنه استكشف كل التوابع المطلوبة.ولم يعمل فقط على إضافة البرنامج الرئيسي وبنود ملف المساعدة التي تظهر في مقطع مجلد التطبيق،ولكن تظهر ثلاث ملفات DLLsإضافية، جميعها يتم استخدامها لتشغيل تقارير المكتبة.كما هو مبين في الشكل التالي.



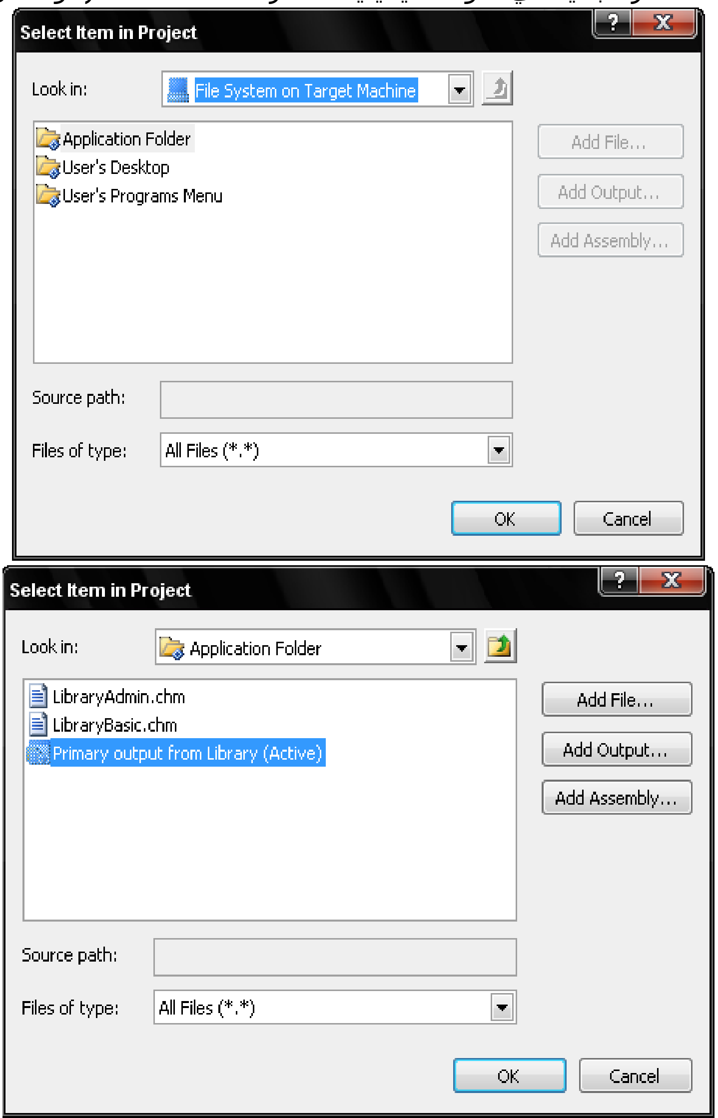
بما أن هذه الـ DLLsالثلاث تم تزويدها من قبل ميكروسوفت كجزء من الدوت نت،فهي لا تعني الكثير لكي أعمل على تخزينها في دليل تنصيب تطبيقي الخاص.فهي ستذهب إلى المخزن الانتقالي للمجمع العام Global Assembly Cache (GAC)،مجلد النظام الخاص الذي يحتفظ بمجمعات الدوت نت المتشاركة.و GACليس واحد من اختيارات المجلد المعروض في المحرر،ولكن يمكن أن يكون كذلك.تأكد من أن محرر نظام الملفات File System Editor على اللوحة اليسارية هو قيد الاختيار(الذي يحوي العنوان File System on Target Machine) ومن ثم استخدم القائمة إجراء Action >>إضافة مجلد خاصAdd Special Folder >>مجلد مخزن انتقالي لمجمع عام Global Assembly Cache Folder.يظهر المجلد الجديد،مجلد المخزن الانتقالي للمجمع العامGlobal Assembly Cache Folder، في اللوحة التي على الجهة اليسارية.اختر بند مجلد التطبيق Application Folder مرةً أخرى،ومن ثم اسحب بنود الـ DLLالثلاث إلى بند مجلد المخزن الانتقالي للمجمع العامGlobal Assembly Cache، كما هو مبين في الشكل التالي.



دعنا نضيف اختصارين إلى نظام المستخدم خلال التنصيب:واحد إلى سطح المكتب والأخر إلى مقطع البرامج في قائمة إبدأ.وكلا الاختصارين يشيران إلى المجمع *Library.exe* الرئيسي.يستبق المعالج السحري للتثبيت حاجاتنا بإضافة المجلد *User's Desktop* والمجلد *User's Programs Menu* إلى محرر نظام الملفاتFile System Editor. كل ما علينا عمله إضافة اختصار إلى كل منهما.

لنبدأ بسطح المكتب.اختر المجلد*User's Desktop* ومن ثم انقر يمين في اللوحة اليمينية(حيث ستظهر الملفات).من قائمة السياق (المنسدلة) اختر إنشاء اختصار جديدCreate New Shortcut. (ونفس الأمر متاح من القائمة "إجراء Action " عندما تكون اللوحة اليمينية قيد

التفعيل).يظهر حوار اختيار بند Select Item في المشروع،كما هو مبين في الشكل التالي،استعرض ضمن بند مجلد التطبيق واختر Primary Output from Library (Active).يظهر الاختصار الجديد في اللوحة اليمينية،منتظراً منك منحه اسم ذو معنى إمنحه الاسم"برنامج المكتبة".

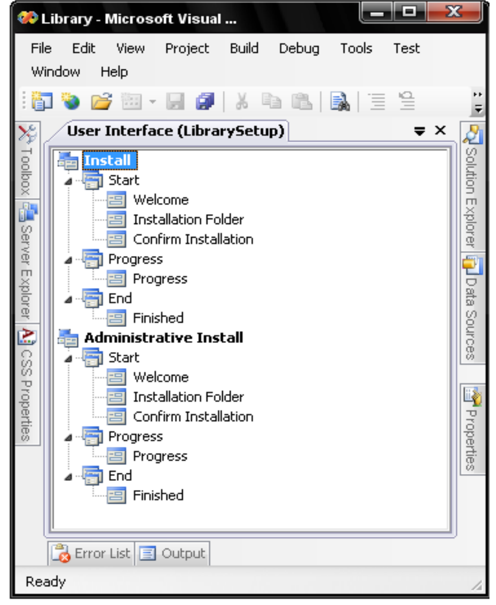


لإنشاء نفس الاختصار إلى قائمة إبدأ،اتبع نفس الخطوات في الفقرة السابقة،ولكن من مجلد *User's Programs Menu* بدل المجلد User's Desktop.

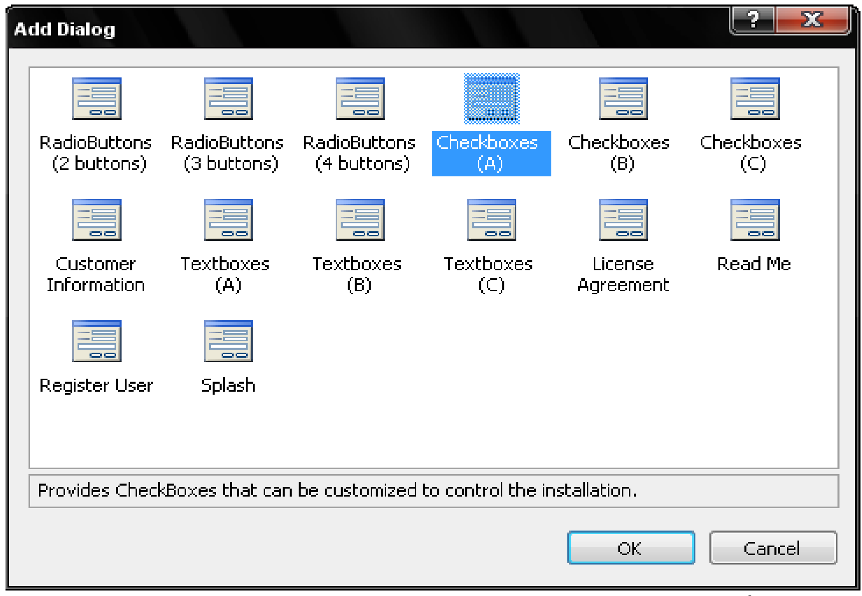
إضافة هذين الاختصارين فكرة جيدة،ولكن عندما أعمل على تنصيب برنامج جديد،أعمل مباشرةً على حذف أي اختصار تم إضافته إلى سطح المكتب.إضافة أيقونة إلى مجلد برامج قائمة إبدأ له معنى أفضل.

ما نحتاجه طريقة لتبديل سلوك برنامج التنصيب بحيث لايعمل على إنشاء أيقونة سطح المكتب إذا كان المستخدم لايريد ذلك.يوفر مشروع التنصيب طريقة لفعل هذا.أولاً،نحتاج إلى إضافة طلب حيث يشير المستخدم إلى أولوية(خيارpreference) أيقونة سطح المكتب،ومن ثم نحتاج إلى التصرف على ذلك الخيار(أو الأولوية).تتضمن الخطوة الأولى تبديل واجهة المستخدم user interface لبرنامج التثبيت.يحدث مثل هذه التغيرات خلال محرر واجهة المستخدمUser Interface Editor . اعرض هذا المحرر من خلال القائمة عرض View >>المحرر Editor >>واجهة المستخدمUser Interface. يظهر محرر واجهة المستخدم،كما هو مبين في الشكل التالي.

إن محرر واجهة المستخدم مقسم إلى نوعي تنصيب رئيسيين: تنصيب Install وتنصيب إداريAdministrative Install. التفرع الإداري يتم استخدامه فقط عندما يريد المدير تخزين صورة التثبيت على مجلد شبكة متشاركة.فلا يسمح لأنواع التغيرات التي نريد عملها.لذلك،دعنا نركز على التفرع تنصيب Installالقياسي، والذي يدير تنصيبات المستخدم القياسية على جهاز العميل.كلا التفرعين



يتضمنان طلبات خطوة –خطوة التي تظهر للمستخدم خلال عمليات ا لتثبيت.تخصيص طلبات تجمع البيانات يمكن أن يتم إضافته فقط إلى مدخلة البدء Startفي التفرع الرئيسي للتنصيبInstall .
خلال التنصيب الحقيقي، تطلب واجهة المستخدم من المستخدم في نمط شبيه بالمعالج السحري.خلال مرحلة استهلال البدءStart ،يجمع البرنامج رغبات المستخدم لباقي المعالجة.حالما ينتهي هذا المقطع،يتواصل التنصيب حتى يكتمل أو يفشل.ما علينا عمله هو إدخال خطوة جديدة في عملية المعالج السحري،تعرض صندوق اختبار checkboxللمستخدم تطلب منه فيما إذا ستظهر أيقونة سطح المكتب أم لا.حقول تجمع بيانات إضافي مثل تلك يمكن أن يتم إضافتها من خلال "الحوارات dialogs" جديدة وهي تحدث لأن تكون حوارات تتضمن صندوق اختبار قابل للتخصيص.في تفرع التنصيبInstall ،انقر يمين على البند بدء Startواختر إضافة حوار Add Dialog من قائمة السياق.تعرض نافذة إضافة حوارAdd Dialog، المبينة في الشكل التالي،الحوارات المتاحة، أختر بند صناديق حوار(A):(A) Checkboxesمن القائمة وانقر موافقOK. يظهر البند الجديد في التنصيب/مقطع البدءInstall/Start. استخدم الفارة واسحبه للأعلى حتى يظهر بين حوار مرحباً Welcomeوحوار مجلد التنصيبInstallation Folder. تتيح لك حوارات صناديق الحوار Checkboxes (A)عرض حتى أربع اختيارات لصناديق اختبار مع عناوين مخصصة.تأكد من أنه تم اختياره في الخطوط الرئيسية للحوار،ومن ثم استخدم لوحة الخاصيات properties لوضع خاصيات هذا الحوار الجديد:
.ضع الخاصية BannerTextإلى "خيارات التنصيبInstallation Options ." وهذا النص يظهر قرب أعلى نافذة الحوار،عارضةً عنوان رئيسي كبير.
.ضع خاصية BodyTextإلى "اختر الخيارات التي ترغب في استخدامها لهذا التنصيب Select the options you wish to use for this installation "



.ضع خاصية Checkbox1Labelإلى"إضافة أيقونة برنامج المكتبة إلى سطح المكتبAdd an icon for ACME Library to the desktop ." وهي تحدد النص الخاص لأداة صندوق الاختبار الأول.
.ضع الخاصية Checkbox1Propertyإلى" LIBRARY_DESKTOP_LINK ."وهذا يمنح صندوق الاختبار اسم نستطيع استخدامه فيما بعد لتبديل عملية التنصيب.
.ضع الخاصية Checkbox1Valueإلى"مختارChecked ."وهي تجعل التنصيب بشكل افتراضي يعمل على إنشاء أيقونة سطح المكتب.
.ضع الخاصيةCheckbox2Visible ، والخاصية Checkbox3Visible،والخاصية Checkbox4Visibleإلى "خطأFalse"، مما يخفي صناديق الاختبار الثلاث الغير مستخدمة.
خلال عملية التثبيت،يرى المستخدم طلب حوار جديد في الشكل التالي.وهو يتضمن نص الرأس banner text، ونص الجسدbody text ، وصندوق حوار مفرد كما تم تركيبه في خاصيات الحوار الخاص.

والآن حان الوقت لاستخدام إعداد صندوق الحوار السابق.أغلق محرر واجهة المستخدم User Interface Editor وارجع إلى محرر نظام الملف File System Editor.اختر مجلد *User's Desktop* في اللوحة اليسارية،ومن ثم اذهب إلى لوحة الخاصيات properties. واحدة من الخاصيات المجدولة هي "الشرط Condition"،والتي تتيح لك تعريف شرط منطقي Boolean حيث عندما يكون صواب true، يعمل على تنصيب الملفات المرافقة على سطح مكتب المستخدم.مهما يكن،إذا كان الشرط "خطأ false"،فلا يتم وضع الملفات المرافقة على سطح مكتب المستخدم خلال التنصيب.ضع هذه الخاصية إلى النص التالي:LIBRARY_DESKTOP_LINK

هذا هو الاسم الذي أعطيناه إلى صندوق الاختبار بالعودة إلى صندوق الحوار المصمم.خلال التنصيب،يختبر برنامج التثبيت اختيار المستخدم، ويبدل تحديث سطح المكتب كما تم طلبه.

شيء أخر لن أعمل على إضافته لإصداري من برنامج التثبيت وهو خط الكود الشامل bar code .

